DM 6.– ÖS 50 SF 6.–

ATARI magazin

1 Jan./Febr. '87

Das unabhängige Magazin für alle Ataris



Editorial



Rainer Grinda

Liebe Leser,

machen Sie sich auch gute Vorsätze für das neue Jahr oder haben Sie das aufgegeben, weil Vorsätze ohnehin nur bis zum 1. Januar halten?

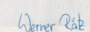
Wir wollen von dieser schönen Regel eine Ausnahme machen. Denn wir haben uns vorgenommen, Ihnen im neuen Jahr mit Atari zu kommen. Und ein Stück weit, sprich mit der ersten Nummer, haben wir unser Vorhaben auch schon in die Tat umgesetzt: Was Sie in Händen halten, ist das neue **ATARI *magazin***. Es kommt aus einem Verlag, der den Anwendern der 8-Bit-Ataris nicht unbekannt ist, denn die Zeitschrift Computer Kontakt begleitet sie seit 3 Jahren mit Informationen und Hilfen für die tägliche Praxis.

Das **ATARI *magazin*** erscheint in Zukunft im Wechsel mit Computer Kontakt als unabhängiges Magazin. Es soll eine lebendige Zeitschrift für alle Atari-Computer werden. Vielleicht haben Sie selbst schon die Erfahrung gemacht, daß sich eine ansprechend gebotene Nachricht leichter aufnehmen läßt. Damit das **ATARI *magazin*** auch in dieser Hinsicht nicht trocken ist, dafür sorgt Rainer Grinda, unser Grafiker. Mit Sprühpistole und guten Ideen gestaltet er die Titel und berät unseren Layouter bei den Innenseiten. Und damit aus dem Ganzen eine runde Sache wird, nämlich ein ansprechendes Heft mit viel Information über die Atari-Computer, dafür sorgt Werner Rätz. Er stellt die Beiträge zusammen und veranlaßt, daß aus Manuskripten Druckseiten werden.

So versuchen wir, unseren Vorsatz fürs neue Jahr zu verwirklichen. Und wenn Sie meinen, das sei uns noch nicht so gut gelungen, dann schreiben Sie uns. Wir werden Ihre Kritikpunkte beherzigen.

Alles Gute im neuen Jahr und viel Spaß bis zum nächsten Heft wünscht

Ihre Redaktion vom
ATARI *magazin*





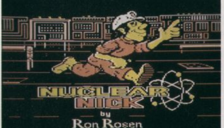
Werner Rätz



Bei Besitzern der Schneider-Computer hat die Firma vortex längst einen guten Namen. Ihre große Stärke sind Speichererweiterungen und Diskettenlaufwerke. Jetzt hat vortex eine Festplatte für den ST fertiggestellt. Unser Mitarbeiter H. H. Fischer hat sich die Hard-Disk vor Ort angesehen.



Computer mit modernen Mikroprozessoren wie der Atari ST sind für Grafikanwendungen besonders geeignet. Aber was taugt der schnellste Computer ohne entsprechendes Programm? Ob das Grafikprogramm "Monostar" dem ST gewachsen ist, lesen Sie in unserem Titelbericht.



Nuclear Nick ist nicht das einzige Spiel, das wir für Sie angeschaut haben. In unseren regelmäßigen Spieletests stellen wir Ihnen Spiele für alle kleinen und großen Ataris vor und hoffen damit, Ihnen die Wahl zu erleichtern.

INHALTSVERZEICHNIS

Das ATARI *magazin* schafft Kontakte!

Wer sich mit seinem Computer beschäftigt, möchte auch Kontakt zu anderen Anwendern. Hier bieten sich die zahlreichen User-Clubs und Benutzergruppen an. Diese Clubanschriften, Neugründungen, Termine, Nachrichten und ander Informationen aus der Szene wollen wir an unsere Leser über diese Kontaktseite weitergeben. Ausführliche Clubvorstellungen sind ebenso möglich, wie Kurzinfos, Änderungen oder Kontaktgesuche.

Wer also einen User-Club leitet oder gründen will, wer Kontakte zu anderen Computerfreaks sucht oder wer besondere Aktivitäten melden kann, sollte uns schreiben.

Unsere Anschrift:
ATARI *magazin*
Postfach 1640, 7518 Bretten

Kiel

Unser Atari-Club sucht noch Mitglieder. Wir sind derzeit 19 User, die entweder einen 600/800XL bzw. einen 130XE besitzen. Bei genügend großer Mitgliederzahl ist eine monatlich erscheinende Club-Zeitung geplant. Ansonsten bieten wir eine Soft- und Hardware-Vermittlung an, auch eine eigene Software-Bibliothek ist vorhanden. Wir klären auch Fragen von Mitgliedern. Info gegen 80 Pf. Rückporto. Anfragen aus dem ganzen Bundesgebiet sind willkommen!

Michael Scholz
Adolfstr. 42
2300 Kiel 1

Vellmar

Unser Club sucht Mitglieder, die Interesse an der Arbeit mit XL/XE (später auch ST) haben. Wir besitzen 800XL, zweimal 1050, 1027 und Panasonic KX-P1080. Touch Tablet sowie eine Programmbibliothek mit 1080 Programmen stehen zur Verfügung. Unser Info erscheint jeden Monat als Clubdiskette zum Preis von 4.– DM. Ein Mitgliedsbeitrag wird nicht erhoben.

Atari-User-Club
Paul Seik
Westring 9
3502 Vellmar

Lippstadt

Wir sind vermutlich ein Club mit den gleichen Interessen wie alle anderen und suchen Gleichgesinnte im Raum Lippstadt, Rietberg, Mastholte und Umgebung. Wir besitzen Atari 400 bis 130XE mit 16K Bibomon, Diskette, Cassette, einen Epson FX80 und eine umfangreiche Bibliothek mit jeder Art von Programmen. Wir suchen keine Leute, die postlagernd arbeiten und nur Programme abstauben wollen. Wir erheben keinen Club-Beitrag und keine Portogebühren. Wir wollen Programme tauschen, Probleme lösen und Tips und Tricks weitergeben. Bei uns gibt es keine Altersgrenze. Für uns sind die Freaks richtig, die mal schnell mit dem Fahrrad oder dem Auto zusammenkommen. (Abholung ist auch möglich.)

Martin Schiller
Schillerstr. 58
4835 Rietberg 3/Mastholte

Hamburg

Wir vom Delmonico User Club Hamburg nehmen noch Mitglieder auf. Wir bieten die Clubzeitschrift "Bitbyter" (vierteljährlich), Public-Domain-Software, clubeigene Software, Kontaktvermittlung und Erfahrungsaustausch. Es wird ein Beitrag von 40.- DM pro Jahr erhoben.

Delmonico User Club Hamburg
Niklas Nebel
Schloßstr. 100
2000 Hamburg 70

Cuxhaven

Die Mitglieder unseres Clubs wohnen fast alle außerhalb von Cuxhaven. Daher veranstalten wir keine Clubtreffen. Wir haben aber gute Beziehungen ins europäische Ausland. Unser Interesse gilt den Atari-8-Bit-Computern. Alle zwei Monate erscheint eine Clubzeitschrift, deren Bezug im Clubbeitrag (10.-DM im Vierteljahr) enthalten ist. Unsere Softwarebibliothek bietet die Möglichkeit, Programme leihweise kostenlos zu beziehen.

Atari-User-Club Cuxhaven
Höfenweg 24
2190 Cuxhaven 1

Ostfriesland

Der ST-User-Club Ostfriesland besteht seit Anfang September 1986. Sein Ziel ist der Informations- und Erfahrungsaustausch zwischen ST-Besitzern. Wir helfen bei Problemen mit Hard- und Software und sammeln sämtliche Public-Domain-Software, die jedem Mitglied kostenlos zur Verfügung steht. Eine eigene Clubzeitschrift ist momentan nicht geplant. Unser Informationsblatt senden wir Ihnen gerne zu. Auch in Aurich gibt es einen ST-Club, mit dem wir in Zukunft zusammenarbeiten werden.

ST-User-Club Ostfriesland
Detlef Koppelkamm
Düsterweg 30
2951 Nortmoor

Waldau

Der Atari-Commodore-Club Waldau sucht Mitglieder, die einen 8-Bit-Atari oder einen Commodore besitzen. Wir bieten ein Clubmagazin (alle zwei Monate), ein vierteljährliches Magazin für Atari XL/XE, eine Programmbibliothek, eine Bastelecke usw. Der Clubbeitrag beläuft sich auf 15.-DM pro Halbjahr.

Atari-Commodore-Club Waldau
c/o Markus Kopp
Angerweg 1
8483 Vohenstrauß 2

Berlin

Bei uns kann jeder mitmachen, der einen Atari mit 8 Bit besitzt. Alle zwei Monate geben wir eine oder zwei Magazin-Disketten heraus und besitzen auch eine Softwarebibliothek. An einer Clubzeitschrift arbeiten wir noch. Der Clubbeitrag beläuft sich auf 10.-DM für zwei Monate.

Atari-Club Berlin
Thomas Gutthat
Schäferstr. 5
1000 Berlin 20

Hallo Atari User!

Wir wollen im Raum GF, WOB, BS und WF einen Computer-Club gründen. Auch eine Club-Bibliothek sowie Club-Treffs mit Unterhaltung und Erfahrungsaustausch sind geplant.

Oliver Hamann
Gladiolenweg 2
3170 Gifhorn

Mein Freund und ich sind beide 14 Jahre alt und suchen Kontakt zu anderen Atari-800-XL-Besitzern mit Datasette oder auch Floppy. Sie sollten in unserem Alter sein und auch im Raum Duisburg wohnen.

Daniel Dommisch
Sandstraße 3
4100 Duisburg 11

Ich suche Kontakt zu anderen Atari-800-XL-Besitzern mit Datasette, die möglichst im Raum HSK oder Korbach-Willingen wohnen sollten.

Heiko Dittkrist
Beethovenstr. 47
5789 Medebach



Diese Farbgrafik läßt von "Karate Kid II" einiges erwarten: schon fast wie Kino aus dem Computer.

Karate Kid II

Eine Vorankündigung der PSL Marketing Ltd. aus England sorgt zur Zeit dafür, daß manche Besitzer eines Atari ST ungeduldig auf die Erscheinung eines Programms warten. Die Rede ist von "Karate Kid II", benannt nach dem erfolgreichen Kinohit.

Eine lauffähige Version des Programms konnte uns PSL noch nicht zur Verfügung stellen, wohl aber eine Demo-Diskette, auf der einige Spielszenen und Hintergrundbilder und ein wenig Sound abgespeichert waren. Wir können also hier noch keinen Testbericht bringen.

Um die ST-Besitzer auf "Karate Kid" einzustimmen, haben wir – auch zur Überbrückung der Wartezeit – einen Screen Shoot ausgesucht, der die fantastische Grafik des Programms zeigt. Mehr von "Karate Kid II", sobald uns das Programm vorliegt.

Rolf Knorre

Atari zwei hoch fünf

Es ist längst kein Gerücht mehr, daß Atari mit einem neuen Rechner auf den Markt kommen will, der auf einer höheren Ebene an den Erfolg des Atari ST anknüpfen soll. Ursprünglich für Ende dieses Jahres geplant, soll nun im Frühjahr 1987, möglicherweise zur Ce-Bit, der neue 32-Bit-Atari TT erscheinen. Das Herz der neuen Maschine wird ein Motorola 68020 sein, getaktet mit 12,5 Mhz. Die Grafikauflösung soll 1024×1024 Punkte betragen.

Der Atari TT wird mit zwei Betriebssystemen ausgestattet sein, mit einem Unix-Abkömmling sowie dem Atari-eigenen TOS. An IBM-Kompatibilität ist laut Atari-Chefentwickler Shiraz Shivji auch beim neuen TT nicht gedacht.

Auch das Preis/Leistungs-Verhältnis wird stimmen. Wie verlautet, soll die Grundkonfiguration nicht mehr als 2500.– bis 3000.– DM kosten.

Spezielles Druckerpapier

Eine minderwertige Hardcopy nach mühevoller Textarbeit wird wohl jeden enttäuschen. Gewöhnliches Druckerpapier hat eine zu grobe Oberfläche, um bei Matrixdruckern einen wirklich sauberen Ausdruck zu liefern. Wesentlich verbessert wird er durch das spezielle Hepa-Schönschreibpapier. Schrift und Grafik werden konturenschärfer und sauberer wiedergegeben.

Auf den ersten Blick ist die feinere und glattere Struktur des Papiers zu erkennen. Dadurch lassen sich über einen Matrixdrucker mit Schönschriftmodus selbst Geschäftsbriefe erstellen, die ein sauberes und klares Erscheinungsbild aufweisen.

250 Blatt im DIN-A4-Format sind zum Preis von 16.90 DM erhältlich.

Hepaversand H. Paust
Parkstraße 18
8000 München 2

Software des Jahres

Bereits seit 3 Jahren wählen Fachjournalisten aus sieben Ländern die Software des Jahres. In den Kategorien Unterhaltung, kommerzielle und technisch-wissenschaftliche Software sowie Software Tools werden Programme ausgesucht, die nach Meinung der Jury hinsichtlich Leistungsfähigkeit und Benutzerfreundlichkeit aus der Masse herausragen.

Gewählt wurden in diesem Jahr im Bereich der Unterhaltung "The Hitchhiker's Guide to the Galaxy", in der Kategorie der kommerziellen Software "Javelin", eine Datenbank mit neuem Konzept. Auf dem technisch-wissenschaftlichen Gebiet wurde das CAD-Programm "Autocad" ausgezeichnet, das, wie "Turbo-Pascal" bei den Tools, nicht zum ersten Mal zur Software des Jahres gewählt wurde.

Frei-Programme

Vor allem aus Amerika stammt eine Vertriebsmethode für Software, die unter dem Namen Shareware, Public Domain oder zu deutsch Frei-Programme auch bei uns bekanntgeworden ist. Programme werden entweder völlig kostenlos abgegeben oder der Benutzer wird gebeten, freiwillig einen vorgeschlagenen Betrag zu bezahlen, wenn er das Programm nützlich findet. Dafür erhält er Updates und weitere Informationen. Dieses Konzept erfreut sich steigender Beliebtheit und Nachfrage, da die Programme oft eine ebenbürtige oder bessere Alternative zu konventionell angebotenen sind. Allerdings konnte es sich, was das Angebot betrifft, in Deutschland noch nicht so recht durchsetzen.

Die Firma EcoSoft, Vertreiber solcher Software, ruft daher die Programmautoren auf, ihre Produkte für diese Vertriebsart zur Verfügung zu stellen. Interessenten wenden sich an:

EcoSoft
Kaiserstraße 21
7890 Waldshut-Tiengen
Telefon 0 77 51 / 79 20

Neues für die Tasche

Bei den kleinsten Computern baut die Firma Sharp ihre führende Stellung weiter aus. Auf der Orgatechnik wurden drei neue Pocket-Computer vorgestellt, von denen zwei speziell auf die Bedürfnisse bestimmter Branchen zugeschnitten sind: Der PC 1425 hat bereits viele Statistikfunktionen integriert, der PC 1262 ist für Banken und Versicherungen gedacht. Das Einsteigermodell (unser Bild) soll zum kleinen Preis an den Computer heranführen. Es besitzt einen Speicher von 2 KByte und 71 Basic-Befehle.

Tschernobyl macht's möglich!

Die Firma Genitron Instrument zeigte auf der Hobby Elektronik 1986 in Stuttgart ein Strahlenmeßgerät zum Anschluß an den Heimcomputer. Eigenen soll es sich sowohl zur ständigen Überwachung der Radioaktivität, etwa in der Umgebung von Atomkraftwerken, als auch zur Untersuchung von radioaktiven Rückständen in Lebensmitteln. Über einen 8-Bit-Userport werden die Daten eingelesen und in Basic weiterverarbeitet.

Brandneu: LDW-Basic-Compiler

15 Monate Entwicklungszeit wurde in den neuen LDW-Basic-Compiler aus den USA investiert. Der amerikanische Softwarehersteller Logical Design Works in Californien will damit alles Dagewesene in den Schatten stellen. Der Compiler ist kompatibel zum ST-Basic, soll aber bis zu 95mal schneller ablaufende Maschinenprogramme erzeugen.

Weitere Leistungsdaten:

- Fließpunktarithmetik mit doppelter Genauigkeit
- Arraygröße unbegrenzt
- Unterstützung von GEMSYS- und VDISYS-Aufrufen
- Keine Zeilennummern erforderlich
- Dateien mit freiem Zugriff
- Programme bis 30000 Zeilen Basic-Code
- CHAIN und COMMON werden unterstützt
- Symbolische Labels sind implementiert
- Compiliergeschwindigkeit: 180 Zeilen/min
- Menügesteuerte GEM-Applikation
- Kein Kopierschutz

Der Compiler wird demnächst auch bei uns verfügbar sein. Ein ausführlicher Testbericht folgt im nächsten **ATARI** ***magazin***.

Space Shuttle

Von der Firma, die zur Zeit noch an "Karate Kid II" arbeitet, haben wir ein weiteres Demo bekommen. Das Programm wird den Titel "Shuttle" oder "Space Shuttle" tragen. Auch hier liegt noch keine lauffähige Version vor. Es soll sich jedenfalls um eine Shuttle-Simulation auf den ST-Computern handeln. Ein Test folgt, sobald das Programm vorliegt.

Rolf Knorre



Für andere Computer ist "Space Shuttle" bereits verfügbar. Jetzt wurde die Weltraumsimulation auch für den ST angekündigt.



20-MByte-Platte

Ein Bericht rund um die neue vortex-Harddisk für den ST

Gute Nerven muß man haben, will man mit Produkten aus dem Hause Atari konkurrieren. Und genau diese scheint die Firma vortex immer noch zu besitzen. Ihr neues Projekt, eine 3,5"-Festplatte mit 20 MByte Kapazität für den Atari ST, dürfte durchaus gute Chancen haben, der Atari-Festplatte Marktanteile abzujagen.

Da Qualität kaum allein durch Werbung an die Abnehmer heranzutragen ist, sind Berichte der Fachzeitschriften oft auslösender Faktor für Erfolg oder Mißerfolg eines Produkts. Das heißt allerdings nicht, daß da irgendwelche Gelder Wunder wirken. Damit kann man höchstens Werbeflächen mieten, in denen man dann fast alles behaupten darf. Ein Bericht entspringt immer noch dem Wunsch, den Lesern gute Produkte vorzustellen, und nicht etwa dem unsinnigen Verlangen, eine Firma zu ruinieren oder in den Himmel zu loben. Summieren sich einmal irgendwelche Negativleistungen zu unerträglichen Rekorden, so kann das schon zu ein paar kritischen, wenn nicht sogar die Käufer warnenden Worten führen. Produziert eine Firma aber ständig Positives, so wird es weitaus schwieriger, dieses auch glaubhaft darzustellen, ohne in den Verdacht eines Marktschreiers zu geraten. Es handelt sich hier also um keinen fairen Vergleich, sondern um die Vorankündigung eines neuen Produkts.

Die preislich äußerst günstige Atari-Harddisk ist sicher keine schlechte Sache. Die Bezeichnung billig wird – weil mißverständlich – bewußt vermieden. Aber wie schon bei den Laufwerken für die CPCs der Firma Schneider, hat vortex die Nase wieder mal ein bißchen weiter vorne. Wie die vortex-Leute es allerdings schaffen, in einer Zeit der ständig wechselnden Computergenerationen so ruhig an die Entwicklung von Hitech heranzugehen, bleibt auch weiterhin ein Geheimnis. Das Qualitätsversprechen des Stempels "Made in Germany" bleibt allem Anschein nach auch bei der 20-MByte-Harddisk gewahrt. Im Moment existieren zwar erst Prototypen, aber die funktionieren schon überzeugend. Lediglich ein paar Kabelbrücken müssen noch ins Platinenlayout übernommen werden. Geplant ist der Start der Auslieferung ohne alle Eile für Mitte Dezember 1986 frei nach dem Motto: kein Fest ohne Festplatte. Aber die Gans, die die Festplatte zieren soll, wird bis zum letzten Tag gemästet. Für den neuen Schneider-PC gibt es die Festplatte schon zum Einstecken in den IBM-Schlitz. Auch die 5.25"-Floppy-Stationen für den Schneider waren Dank ihrer Zuverlässigkeit ein voller Erfolg und ein erster Test für das vortex-Knowhow.

Geboten wird dabei so einiges, von den 20 MByte des zusätzlichen freien Speichers mal ganz abgesehen, denn allzuviel ist das ja auch wieder nicht, eher die unterste Grenze. 10-MByte-Platten sind noch nichts Ganzes, aber schon nichts Halbes mehr. Der eine oder andere Computerbesitzer oder EDV-Beschäftigte dürfte wohl schon von Größenangaben um die 300 MByte gehört haben, die bei den CD-ROMs zu erwarten sind. Atari will da ja logischerweise auch mitmischen. Daß es auch noch größere Datenbanken gibt, soll hier nicht interessieren.

Was also die Kapazität betrifft, dürften die 20 MByte für viele Anwendungen vollkommen ausreichen. Vor allem kleine und mittlere Betriebe werden hier genügend Platz für ihre Kundenstammdaten und ihre Lagerverwaltung finden. Und obwohl es sich bei den STs sicher um astreine Anwendermaschinen handelt, so verfügen sie doch über hervorragende Eigenschaften zur Spieleprogrammierung. Was anderes als eine Anwendung ist eigentlich ein Spiel? Auf 20 MByte lassen sich ganz schön viele Bilder oder Sprites und anderes Zubehör unterbringen. Wozu noch mühsam Grafik entwerfen? Szenario und Figuren einfach mit der Videokamera aufnehmen, digitalisieren, das übliche Maschinenprogramm dazu, alles auf ein paar Disketten für den Versand verpackt und beim Endverbraucher mit einem Kopierprogramm einmal auf Harddisk überspielt, würde der Unterhaltungselektronik zu mehr Ansehen verhelfen.

Allein schon der Aufbau der vortex-Festplattenstation ist beeindruckend. Das schlichte, aber solide Stahlblechgehäuse ist in mattem Atari-Grau lackiert – die exakte Farbmischung scheint nun gefunden und umschließt sicher Festplatte, Controller- und Adapterkarte sowie das Netzteil. Durch die kompakte Bauweise konnten die Maße des 3.5"-vortex-Floppylaufwerks beibehalten werden. Dies entspricht in der Länge etwa einer Stange Zigaretten (die Harddisk kostet zwar mehr Geld, schadet der Gesundheit aber weniger) und im Umfang einer Musikcassettenhülle. Kurz gesagt, die Station benötigt kaum Platz.

Das Gehäuse selbst ist zwar ein bißchen länger geworden, dafür fehlen aber die Kühlrippen, die bei der Floppystation an der Rückwand befestigt sind. Diese wurden durch ein beinahe geräuschloses Gebläse ersetzt, welches im Netzteilblock integriert werden konnte. Man muß sein Ohr schon dicht an die geschützte Ausblasöffnung halten, um ein deutliches Geräusch wahrnehmen zu können. Die Menge der ausgestoßenen Luft läßt allerdings eine gute Kühlung vermuten. Die geräuschlose und gut geleitete Luftumwälzung kann ruhig als ein Kriterium für Kaufentscheidungen herangezogen werden.

Die bei Festplatten gewohnte schwarze Kunststofffront mit Schlitzen für den Lufteinlaß entfällt. Die Luft wird nunmehr durch eine geeignete Öffnung am vorderen Ende des Gehäusebodens angesaugt. Dank der vier Gummifüße ist für ausreichende Bodenfreiheit gesorgt. Die Front selbst wird nur noch durch die Power-on-Leuchtdiode durchbrochen. Verwechslungen mit einem Floppylaufwerk dürften da wohl kaum vorkommen.

Für den eigentlichen Betrieb ist dies jedoch nicht entscheidend. Da zählt schon eher die geringe Leistungsaufnahme und die volle Kompatibilität zur Atari-Harddiskstation. Man muß diese also nicht zähneknirschend verkaufen. Vielmehr ist eine Backup-Harddisk für wichtige Geschäftsdaten durchaus von Nutzen, wenn auch ungewöhnlich. Tapestreamer sind schließlich auch nicht gerade billig und bei der täglichen Arbeit überhaupt nicht nutzbar. Lediglich bei den Datenträgern lassen sich da ein paar Mark sparen, die aber durch einen eventuellen hohen Zeitverlust meist wieder abgegolten werden. Zudem bieten sich Disketten-Backups an. Bei Kapazitäten von 1 MByte und mehr halten sich die Kosten durchaus in Grenzen. 20 Disketten sind sicher noch im Budget enthalten.

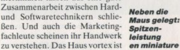
Auf der in ihren Ausmaßen wirklich winzigen Festplatte, die so klein sein muß, damit sie auch auf eine Slotkarte für IBM und Kompatible paßt, ist die Controllerkarte angebracht. Sie läßt einen klaren Aufbau erkennen. Über dieser wiederum sitzt noch die Adapterkarte, die die Kommunikation mit der DMA-Schnittstelle des Computers ermöglicht. Auch diese ähnelt durch ihre aufgeräumte Architektur einem Villen-Vorort irgendwo in Amerika. Mit drei Dip-Schaltern kann der Adapterkarte eine Device-Nummer von null bis sieben zugeordnet werden. Dadurch ist der Anschluß und der Betrieb von bis zu acht Festplatten gleichzeitig möglich. Denkbar wäre dann ein Festplattenarchiv anstelle der bisher üblichen Diskettenboxen. Die 20-MByte-Station selbst ist einfach und schnell mittels des zum Lieferumfang gehörenden Kabels an der DMA-Buchse eines Atari-Computers angeschlossen. Einfach einstekken, fertig!

Die auf 3,5"-Diskette mitgelieferte Treibersoftware kopiert man sich am besten in einen Auto-Ordner, dann wird die Festplatte nach dem Einschalten oder einem Reset des ST gleich eingebunden. Alle notwendigen Parameter sind nach der Erstinstallation auf dem Bootsektor der Platte enthalten. Man kann die Festplatte in bis zu vier logische Laufwerke unterteilen, was die Übersichtlichkeit unbedingt erhöht. Auf dem Desktop erscheint dann für jedes dieser Laufwerke ein Icon mit der Unterschrift Harddisk und einem Kennbuchstaben.

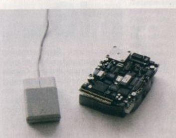
Weiter vorne im Text wurde schon einmal angesprochen, die eventuell schon vorhandene Atari-Harddisk als Backup zu verwenden. Denn in Punkto Geschwindigkeit will vortex einfach an das Maximum heran. Sage und schreibe 970 KByte pro Sekunde werden da von oder zum Computer geschaufelt. Der DMA-Baustein schafft gerade 1 MByte pro Sekunde.

Aufgezeichnet wird im RLL2.7-Format. Die mittlere Zugriffszeit der vortex-Station liegt bei 80 Millisekunden. Egal wo der Schreib-/Lesekopf gerade steht, die Logik benötigt durchschnittlich 80ms, bis sie ihn an der erforderlichen Stelle positioniert hat.

Produkte mit diesen Qualitätsmerkmalen lassen auf eine gute Zusammenarbeit zwischen Hard- und Softwaretechnikern schließen. Und auch die Marketingfachleute scheinen ihr Handwerk zu verstehen. Das Haus vortex ist



Neben die Maus gelegt: Spitzenleistung en miniature



unbedingt zu den Firmen zu zählen, die das Image der Computerbranche tüchtig aufpolieren und dies nicht nur im Inland. Auch Franzosen und Engländer wissen inzwischen das schwäbische Knowhow zu schätzen.

Zieht man einmal die wirklich nützliche MS-DOS- oder auch CP/M-Software zum Vergleich heran, so dürfte dem ST und allem Funktionstüchtigem, was mit ihm zusammenhängt, wohl eine brauchbare Zukunft beschieden sein. Das Image des ST wird vom blitzgescheiten Außenseiter zum Minimalstandard hinüberwechseln.

H. H. Fischer

Praktische Hilfen

Der Umgang mit den Speichermedien RAM-Disk und Harddisk wird mit diesen Programmen erleichtert

Die Begriffe RAM-Disk und Harddisk tauchen in Zusammenhang mit Computern immer wieder auf. Durch die hervorragende Hardware-Ausstattung der ST-Computer werden ihre Benutzer besonders häufig damit konfrontiert. Ich will hier nicht alle Möglichkeiten

16 Bit

dieser beiden Speichermedien ausleuchten und auch keine technischen Details bringen. Gerade Computer-Neulinge werden aber natürlich nicht genau wissen, was mit obigen Begriffen gemeint ist. Für diese Leser möchte ich doch kurz eine grobe Erläuterung geben.

Bekanntlich benötigt man für ein Laufwerk Disketten zur Datenspeicherung (bei Atari im Format 3,5"; andere Formate sind z.B. 3" für den CPC und den Joyce von Schneider und 5,25" für IBM – und kompatible Computer bzw. Laufwerke). Nach Formatierung kann die Diskette Datenmengen von 360 bis 720 KByte (und mehr) fassen. Für den normalen Heimanwender reicht diese Speicherkapazität in der Regel völlig aus.

Anders sieht es aus, wenn man den ST auch beruflich einsetzt. Wer damit z.B. Buchhaltung macht oder eine Datenbank zur Lagerhaltung oder Adressenspeicherung anlegt, sieht sich unter Umständen zu häufigem Diskettenwechsel gezwungen.

Hier kann die Harddisk Abhilfe schaffen. Für die ST-Computer gibt es mittlerweile verschiedene Ausführungen von 10 bis weit über 100 MByte. Die Harddisk ist vom Aufbau her einem normalen Diskettenlaufwerk vergleichbar. Allerdings können hier keine Disketten eingelegt werden. Im Innern des Gehäuses rotiert eine sogenannte Festplatte, die vom Anwender nicht entfernt werden kann. Da die Harddisk mit wesentlich höherer Geschwindigkeit arbeitet als ein normales Laufwerk und die Daten dichter gepackt werden, sind sehr hohe Speicherkapazitäten möglich.

Der Umgang mit einer Harddisk ist eigentlich sehr einfach. Sie wird wie ein normales Laufwerk angesprochen, kann also zum Speichern und Laden von Daten verwendet werden. Das Laden erfolgt schneller als beim Diskettenlaufwerk. Trotzdem muß der Anwender einige Dinge beachten, die ihm vielleicht neu sind.

Unterstützung bietet das Programm "Harddisk Help" der Firma G DATA, das zwei Hauptfunktionen besitzt. Die Funktion TREE dient dem Anlegen und Ausgeben eines übersichtlichen Inhaltsverzeichnisses der Harddisk. Man kann sich leicht vorstellen, daß bei einer Datenmenge von vielleicht 20 MByte der Überblick schnell verlorengehen kann. Die Ausgabe des Inhaltsverzeichnisses kann über Bildschirm oder Drucker erfolgen.

Wesentlich wichtiger ist die zweite Funktion, die die Bezeichnung BACKUP trägt. Bei normalem Einsatz ist die Harddisk unproblematisch. Trotzdem kann es zu dem berüchtigten Headcrash kommen. Da Festplatte und Schreib-/Leseköpfe mit enormer Geschwindigkeit arbeiten, ohne sich zu berühren, kann ein Stoß an den Arbeitstisch einen solchen Crash verursachen. Ganz so schnell passiert das zwar nicht, möglich ist es aber. Alle gespeicherten Daten können dabei verlorengehen. Die Anlage einer Back-Up-Kopie des Platteninhalts ist deshalb äußerst ratsam. In der Industrie werden diese normalerweise mit Bandlaufwerken, genannt Streamer, erstellt. Die Anschaffung eines Streamers ist aber recht kostspielig.

Da jeder ST-Benutzer bereits ein Diskettenlaufwerk hat, wäre es naheliegend, damit eine Kopie anzufertigen. "Harddisk Help" unterstützt alle Aktionen zwischen Harddisk und Laufwerk. Das beginnt bei der Angabe, wie viele Leerdisketten nötig sind. Die Disketten können auch in einem Spezialformat formatiert werden. Das Überspielen der Daten bereitet keine Schwierigkeiten. Alle Aktionen werden am Monitor kommentiert. Das gilt auch für den umgekehrten Weg, also das Laden der Daten von Diskette in die Harddisk. Das Programm "Harddisk Help" kann auf die Harddisk übertragen werden; auf einen Kopierschutz wurde hier verzichtet. Kein Harddisk-Betreiber sollte auf dieses Hilfsprogramm verzichten.

Nach diesem Ausflug zu den teuren Speichermedien (eine 20-MByte-Harddisk für den Atari kostet ca. 2.500,– DM) wollen wir wieder in Bereiche zurückkehren, die jedem ST-Anwender zur Verfügung stehen. Gemeint ist die RAM-Disk. Darunter versteht man den Einsatz des internen RAM als Speichermedium. Auch diese Technik ist leicht zu bedienen, wenn Sie einmal installiert ist. Dabei hilft das Programm "G Ramdisk". Besonders Besitzer eines Mega Atari wie des 1040 ST mit nur einem Diskettenlaufwerk werden mit der RAM-Disk viel Freude haben.

Was ist zu tun? Zunächst sollte das Programm auf die eigene Startdiskette übertragen werden. Es handelt sich nämlich um ein Accessory, das beim Systemstart gebootet werden kann und dann ständig im Desktop zur Verfügung steht. Zu diesem Zweck ist auf der Originaldiskette gleich ein Programm installiert, mit dem 2 Kopien angefertigt werden können.

Der nächste Schritt ist das Anmelden einer neuen Diskettenstation, was auch vom Desktop aus erledigt werden kann. Die neue Station sollte die Bezeichnung D RAM-Disk tragen. Sobald ihr Symbol auf dem Monitor erscheint, sollte man mit ARBEIT SICHERN dafür sorgen, daß dieses auch bei einem erneuten Systemstart sofort zur Verfügung steht.

Jetzt kann das Programm "G Ramdisk" aufgerufen werden. Auf dem Monitor erscheint ein Formular, das auszufüllen ist ("G Ramdisk" läuft auch mit einem Farbmonitor, allerdings ist hier die Bildschirmdarstellung nicht mehr optimal). Im Feld links oben wird die Größe der RAM-Disk festgelegt. Hier kann der Anwender je nach freiem Speicher und benötigtem Speicherplatz einen Wert eingeben. Wer nun einen ST mit "nur" 512 KByte besitzt, muß natürlich Abstriche machen. Beim Mega ST könnte man z.B. eine RAM-Disk mit rund 360 KByte dimensionieren, um beim Anfertigen einer Sicherheitskopie (einer einseitigen Diskette) nicht ständig die Disketten wechseln zu müssen. Glücklicherweise kann man die Speichergröße der RAM-Disk unter normalen Umständen jederzeit verändern, also vergrößern oder verkleinern oder auch ganz abschalten. Mir ist es schon passiert, daß ich ein Programm nicht mehr laden konnte, weil der freie Speicher zu klein war (der Platz für die RAM-Disk steht ja nicht mehr als freier Speicher zur Verfügung).



Das Eröffnungsmenü

Unter dem Feld für die Disk-Kapazität befindet sich ein weiteres Eingabefeld. Hier kann ein Druckerpuffer angelegt werden. Dies wird besonders Anwender begeistern, die viel zu Papier bringen. Normalerweise muß man ja beim Ausdruck warten, bis der Drucker fertig ist, bevor man weiterarbeiten kann. Durch den Puffer entfällt diese Wartezeit, da der auszudruckende Text dort zwischengespeichert wird.

Auf der rechten Seite des Formulars kann man weitere Eingaben machen. Hier wird z.B. die RAM-Disk mit einem Kennbuchstaben versehen. Außerdem kann man festlegen, ob sie normal oder resident angelegt wird. Bei der residenten Art übersteht ihr Inhalt einen RESET und unter Umständen auch einen Programmabsturz. Vor dem endgültigen Abschalten des Computers muß der RAM-Disk-Inhalt natürlich noch auf Diskette kopiert werden.

Die Möglichkeiten, die eine RAM-Disk bietet, sind vielfältig. Vorrangig sind natürlich alle angesprochen, die nur eine Diskettenstation besitzen. Gerade beim Kopieren von Files oder Disketten kann man viel Zeit und Nerven sparen, weil der Diskettenwechsel entfällt. Da Programme, die in der RAM-Disk abgespeichert sind, sehr schnell geladen werden, könnte man auch verschiedene Programme und Teile davon dort ablegen, die man während der Arbeit benötigt.

Wie "Harddisk Help" ist auch "G Ramdisk" eine fantastische Hilfe im täglichen Umgang mit dem ST. Wenn man einmal mit diesen Programmen gearbeitet hat, wird man kaum noch auf sie verzichten wollen.

System: Atari ST
Hersteller/Bezugsquelle:
G DATA, Bochum

Rolf Knorre

Bei den riesigen Datenmengen auf einer Festplatte darf die Übersicht nicht verlorengehen. Hard Disk Help hilft!

Zeichenprogramm der Superlative

Schon mit der Schwarz-Weiß-Version dieses Programms zeigt der ST seine Muskeln.

Die enormen Grafikfähigkeiten der ST-Computer haben dazu beigetragen, daß ständig neue Zeichenprogramme auf den Markt kommen. Dabei ist diese Bezeichnung in vielen Fällen gar nicht ausreichend. Bildgestaltung, Grafikverarbeitung oder kreatives Hilfsmittel wären hier schon besser angebracht.

16 Bit

Aus der Vielzahl der vorhandenen Software ragt meiner Meinung nach ein Programm besonders heraus. Gemeint ist "Monostar". Der erste positive Punkt ist, daß das Programm zu 90% in GfA-Basic geschrieben wurde. Dies ist dem Vorwort der Anleitung zu entnehmen, bei der Arbeit mit "Monostar" kann man es nicht merken. Damit ist das Programm ein hervorragendes Beispiel für die Leistungsfähigkeit des GfA-Basic, die ja schon länger bekannt ist, aber noch nie so gut demonstriert wurde.



Grafik fast zum Reinbeißen: Der Apfel wird gleich mitgeliefert

Wichtiger noch ist aber die Leistungsfähigkeit des Programms selbst. Schon das Einarbeiten macht viel Spaß. Da "Monostar" voll unter GEM läuft, wird die Tastatur nur in Ausnahmefällen benötigt. Fast alle Operationen können mit der Maus durchgeführt werden. Da man kaum etwas falsch machen kann, genügt es, alle Optionen auf gut Glück auszuprobieren. Das gut gemachte Handbuch muß man nur in speziellen Fällen zu Rate ziehen.

Als Grundstock bietet "Monostar" alle Funktionen, die man von einem Zeichenprogramm erwartet. In Stichworten wären das:

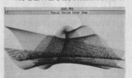
- Freihandzeichnen
- Punkte setzen
- Striche und Linien
- Strahlen
- Vielecke
- Rechtecke und Quadrate
- Kreise und Ellipsen

Eingesetzt werden all diese Optionen nach einem immer wiederkehrenden Muster: Anklikken der Option, rechte Maustaste drücken (die Menüleiste am oberen Bildrand verschwindet) und mit einem Druck auf die linke Taste die Option ausführen. Erneutes Betätigen der rechten Maustaste ruft die Menüleiste zurück. Nach sehr kurzer Einarbeitungszeit klappt dieser Ablauf reibungslos.

Schon die Standardoptionen bieten mehr, als das bei anderen Programmen der Fall ist. Man kann z. B. wählen, ob Rechtecke, Kreise usw. nach dem Zeichnen direkt gefüllt werden oder nicht. Beim Freihandzeichen hat man eine Möglichkeit, die ich bisher überhaupt noch nicht gesehen habe. Sie trägt die Bezeichnung GLÄTTE. Alles was man von Hand auf den Bildschirm malt, wird nachträglich vom Programm geglättet. Besonders hilfreich ist diese Option für Leute, die keine ruhige Hand haben. Kleine Fehler in der Zeichnung werden korrigiert. Wer diese aus künstlerischen Erwägungen heraus gerne belassen möchte, kann die Option auch abschalten.

Kommen wir jetzt zu weiteren Besonderheiten, die "Monostar" bietet:

- BEWEGE: Ein beliebiger Bildschirmbereich kann ausgeschnitten und verschoben werden.
- KOPIERE 1 MAL: Arbeitet wie BEWEGE, schneidet aber



nicht aus, sondern kopiert an einen beliebigen Bildschirmbereich.
- KOPIERE X MAL: Wie vorhergehende Funktion, jedoch wird bei jedem Mausklick eine Kopie erstellt (beliebig oft anwendbar).
- SPRÜHE: Ruft den beliebten Spraydoseneffekt auf. Art und Intensität des Sprühens sind einstellbar.
- DEHNE/STAUCHE: Ein beliebiger Bildschirmausschnitt kann horizontal oder vertikal gedehnt oder zusammengedrückt werden. Damit kann man Bilder vergrößern oder verkleinern.
- BIEGE: Diese Funktion dient dem Verformen der Bilder. Hierzu sind Voreinstellungen



möglich. Wahlweise kann man rechts unten oder links oben biegen, knicken oder auch freihändig manipulieren. Die Ergebnisse sind wirklich sehenswert.
- SPIEGLE: Ein beliebiger Ausschnitt kann horizontal oder vertikal gespiegelt werden.
- INVERTIERE: Jeder schwarze Punkt wird in einen weißen umgewandelt und umgekehrt.
- LUPE: Ein kleiner Bildausschnitt (kann ausgewählt werden) wird stark vergrößert angezeigt, was eine enorme Hilfestellung für Feinarbeiten bietet.

Diese Optionen sind zum Teil einmalig. Auch die Ausführungsgeschwindigkeit ist beeindrukkend. Neben den bisher aufgeführten Funktionen sind auch zahlreiche Voreinstellungen möglich, die das Spektrum von "Monostar" noch mehr erweitern. Ohne ins Detail zu gehen, möchte ich hier die verschiedenen Füllmuster sowie Text- und Stricharten erwähnen. Mit der Verknüpfungs-Option läßt sich einstellen, ob beim Kopieren ein Objekt fest oder durchsichtig dargestellt wird. Diese Aufzählung ist aber noch lange nicht vollständig.

Die Arbeit mit "Monostar" ist auf zwei Bildschirmen möglich; ein weiterer kann zur Abspeicherung der Spezialarbeiten wie Biegen usw. genutzt werden. Den zu bearbeitenden Bildteil muß man dann dem Hauptbild entnehmen und auf die Extraseite kopieren. Wenn keine Spezialarbeiten vorgenommen werden, kann man auch drei Bilder gleichzeitig im Speicher behalten und bearbeiten.

Neben dem normalen Format wird auch DIN-A4-Größe bereitgestellt; hier muß die Gestaltung dann aber in 2 Teilen erfolgen. Bilder, die mit dem "Profi Painter" gezeichnet wurden, können so mit "Monostar" weiterverarbeitet werden. Überhaupt ist die Ladeoption nicht starr geregelt. Man kann zwar nicht sämtliche Grafikprogramme laden, die Möglichkeiten sind aber doch vielfältig.

"Riesige Datenmengen werden bewegt"

Interessant ist auch die Option Objekte abspeichern. Damit sind einzelne Bildelemente gemeint, von denen die "Monostar"-Diskette bereits einige enthält.

Selbstverständlich können die erstellten Kunstwerke auch zu Papier gebracht werden, und zwar mit einem Epson oder kompatiblen Druckern. Damit ist ein Ausdruck des ganzen Bildes (einschl. DIN A4) oder eines Teils davon möglich.

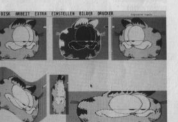
Es gibt eigentlich nur einen Kritikpunkt, das Fehlen einer Zeichenhilfe in Form eines Lineals oder eines Rasters, das über den Arbeits-Screen gelegt werden kann. Damit könnte man sich beim Zeichnen erheblich besser orientieren. Vielleicht kann der Hersteller das in einer zukünftigen Version berücksichtigen. Ansonsten ist "Monostar" ein hervorragendes Programm, das den meisten Anwendern gute Dienste leisten wird und seinen Preis durchaus rechtfertigt.

Zum Abschluß noch ein Wort zu den Geräten. Benötigt wird ein Monochrom-Monitor, farbige Grafiken sind nicht möglich. Es soll aber bereits "Colorstar" geben, das ich nach Erhalt ebenfalls gerne vorstellen werde. Zum Speicherplatz wäre zu sagen, daß "Monostar" selbst nicht sehr lang ist, im Betrieb mit den 3 Screens aber rund 300 KByte benötigt. Bei den Optionen BIEGE und DEHNE/STAUCHE werden unter Umständen sogar rund 500 KByte belegt. Man sollte also schon einen ST mit 512 KByte und eingebauten ROMs oder einen Mega ST besitzen, um alle Optionen einsetzen zu können. (Reicht der Speicherplatz bei einer Spezialoption nicht aus, erhält man vom Programm eine entsprechende Mitteilung mit der Angabe, wieviel Prozent des Bildes noch bearbeitet werden können.)

System: Atari ST, monochrom
Hersteller/Bezugsquelle: Brain Wave Wuppertal
Preis: ca. 99.– DM

Rolf Knorre

Die Optionen sind vielfältig. Kater Garfield in allen Variationen.



Der Champion

Das Beste, was derzeit an Schachprogrammen zu haben ist: Psion Chess. Das Programm wurde jetzt an den Atari ST angepaßt.

Das erste erwähnenswerte Schachprogramm für den Atari ST ist Psion Chess, Gewinner der Schachweltmeisterschaft der Mikrocomputer. Dieses Programm gibt es nun für das neue Gerät mit 16 Bit. Ursprünglich für die PCs von IBM, kompatible sowie Apricot-

16 Bit

Rechner entwickelt, ist die Umsetzung auf den ST wirklich hervorragend gelungen. Die grafischen Möglichkeiten und die Rechenleistung der neuen Generation wurden voll genutzt.

Zunächst besticht die dreidimensionale Darstellung des Schachbretts, die vor allem auf dem Monochrom-Monitor eine verblüffende Qualität erreicht. Wer sich nun näher mit dem Programm auseinandersetzen möchte und das Handbuch aufschlägt, wird nur zwei Seiten einer Beschreibung vorfinden. Die restlichen Erklärungen sind im Programm untergebracht. Schnelles Nachschlagen und ruhiges Durchlesen sind kaum möglich. Zunächst muß man die Hilfsoption anwählen, den gewünschten Punkt anklicken, auf das Nachladen von der Diskette warten und sich durch eine sehr langsam scrollende Hilfsseite durcharbeiten. Warum die Beschreibung nicht im Handbuch ausführlicher geliefert wird, ist mir unverständlich. Diese Form der Anleitung kann wirklich keinem ehrlichen Käufer zugemutet werden; Raubkopierern wird das aber gerade recht sein. Man muß dem Programm allerdings zugute halten, daß die Anleitung in sechs Sprachen, darunter auch Deutsch, gewählt werden kann. Sogar die Menütexte werden eingedeutscht und erscheinen nach Speichern des entsprechenden Parameters auf Diskette auch in Zukunft so.

Die Spielstärke von "Psion Chess" übertrifft die der meisten Schachprogramme auf Mikrocomputern. Das garantiert nicht zuletzt der Titel bei der Schachweltmeisterschaft. Insgesamt gibt es 28 Schwierigkeitsgrade, die sowohl den Erfordernissen des Anfängers als auch des erfahrenen Turnierspielers gerecht werden. Auf der letzten Stufe läßt sich der Computer mit der Auswahl des optimalen Zuges so lange Zeit, bis der Spieler ihm Einhalt gebietet (eine hervorragende Methode, um Briefschachpartner zu schocken). Normalerweise rechnet "Psion Chess" auch während der Bedenkzeit des Gegners, doch auf Wunsch kann man diesen kleinen Vorteil auch noch abschalten.

Das Spielen selbst gestaltet sich recht komfortabel. Man klickt einfach die gewünschte Figur an und setzt sie auf das entsprechende Feld. Alle Veränderungen in Parametern und Funktionen werden über Pull-Down-Menüs vorgenommen. Die Darstellung ist sowohl im attraktiven 3-D- als auch im nützlichen 2-D-Modus möglich, bei dem die Notation aller Züge parallel neben dem Brett sichtbar ist. Natürlich darf man mit allen Annehmlichkeiten des Computerschach rechnen: Zug zurücknehmen, Seitenwechsel während des Spiels, Vorschlag des besten Zuges durch den Rechner und Einblick in die Spielanalyse. Letzteres ermöglicht es dem Spieler, die Arbeitsweise des Programms besser zu verstehen. Es wird angezeigt, wie der Computer die Situation bewertet und welche Züge er in Erwägung zieht. In einer sehr komfortablen Option können Spielsituationen konstruiert und dann weitergespielt werden. Auch Schachprobleme mit Matt in bis zu 8 Zügen bereiten dem Programm keine Probleme.



Nicht nur Spielstärke, sondern auch die Grafik zeichnen ein gutes Schachprogramm aus

In der Regel wird man alleine gegen den Rechner spielen. Es gibt aber noch zwei weitere Modi, in denen man den Computer zur Demonstration gegen sich selbst antreten oder als Schachbrett fungieren lassen kann, wenn man gegen einen menschlichen Gegner spielt. Dabei kann man auf Wunsch alle Züge oder auch eine interessante Spielsituation per Hardcopy ausdrucken lassen.

BERICHT



Ein toleranter Gegner: Psion Chess läßt sich seinem Gegner anpassen

Eine der interessantesten Optionen ist die Nachspielfunktion von 50 Meisterpartien aus über 150 Jahren Schachgeschichte. Sie sind auf Diskette gespeichert und können in einem speziellen Modus nachgespielt werden. Das kann auf Wunsch Zug für Zug, aber auch automatisch geschehen. Wer es sich zutraut, kann auch in das Geschehen eingreifen und weiterspielen.

"Psion Chess" ist das mit Abstand beste Schachprogramm für

"150 Jahre Schachgeschichte auf Diskette"

den Atari ST. Die zukünftigen Konkurrenten werden es schwer haben, gegen diese solide Mischung aus hervorragender Aufmachung und großer Spielstärke zu bestehen. Jedem Schachfreund kann es sehr empfohlen werden.

System: Atari ST (512 KByte, SW/Farb-Monitor)
Hersteller: Psion Ltd.
Bezugsquelle:
Profisoft GmbH, Osnabrück

Thomas Kern

8-Bit-Ware

Software und Bücher für die kleinen Ataris

Die 8-Bit-Maschinen von Atari sind außerordentlich leistungsstarke Computer in Punkto Grafik und Sound. Nur leider war gute und gleichzeitig preiswerte Software bisher noch mit der Lupe zu suchen. Wir haben daher einige interessante Software-Pakete und Bücher für Sie zusammengestellt.

Software

Zuerst zu Grafik und Sound. Die "Soundmachine" ist ein vierstimmiger Synthesizer, mit dem sich alle Klangmöglichkeiten des Atari-Computers ausnutzen lassen. Die Noten werden per Joystick oder Tastatur in vier Notenzeilen eingetragen und können per Tastendruck gespielt werden. Darüber hinaus lassen sich die Musikstücke dank des eingebauten Musik-Compilers auch in eigenen Programmen verwenden. Zehn verschiedene Hüllkurven sorgen für den richtigen Sound, sogar Schlagzeugeffekte sind möglich.

Mit dem "Design-Master" können Sie Ihren Atari-Computer in ein CAD-System verwandeln. Das Programm arbeitet in der höchsten Auflösung des Computers und eignet sich daher für Schaltpläne, Diagramme, Grundrisse oder sonstige Zeichnungen. Alle üblichen Zeichenfunktionen stehen zur Verfügung: Freihand, Linien, Rechtecke, Kreise und Füllfunktion. Daneben hat "Design-Master" aber noch einige Spezialitäten zu bieten: Zwei unabhängige Bildschirme, eine Kopierfunktion für Bildausschnitte (auch von einem Bild ins andere), über 100 verschiedene Schriftarten, ein einblendbares Gitter mit Maßstab, ein großes Fadenkreuz als Cursor und schließlich eine Zoomfunktion mit Scrolling. Ein weiterer Programmteil erlaubt das Ausdrucken der Zeichnungen. Die neueste Version (1.2) des Hardcopy-Programms unterstützt 7-Nadel-Drucker wie den Atari 1029 oder Seikosha GP 100AT und ermöglicht auf diesen Druckern einen DIN A4 (!) großen Ausdruck. Selbstverständlich sind nach wie vor auch 8- und 9-Nadel-Drucker wie Epson und Kompatible verwendbar.

Auch bei den Programmier-Werkzeugen gibt es einige nützliche Programme. Wenn Sie Basic mit Maschinenroutinen unterstützen möchten, dann bietet Ihnen "Monitor XL" eine unersetzliche Hilfe. Mit diesem Basic-Zusatz können Sie Maschinenprogramme von Diskette laden, auflisten (disassemblieren) und verändern. Zur Fehlersuche können Maschinenprogramme in Einzelschritten ausgeführt werden. Auch das gezielte Laden und Speichern von einzelnen Disk-Sektoren ist mit "Monitor XL" kein Problem.

Zur komfortablen Entwicklung von Maschinenprogrammen dient der "ATMAS-II"-Makroassembler. Die Programme werden mit einem komfortablen Full-Screen-Editor eingegeben und vom Assembler in Sekundenschnelle in ausführbare Maschinenprogramme übersetzt. Die Makrofähigkeit von "ATMAS-II" erleichtert die Programmierung, besonders da auf der Diskette einige Makros wie PLOT, DRAWTO oder OPEN bereits mitgeliefert werden. Weitere Makros befinden sich auf der Zusatzdiskette (ATMAS-Toolbox). Sie enthält Makros zur Ein- und Ausgabe von Zahlen und Texten, zum Rechnen mit 16-Bit-Integers sowie einige nützliche Utilities wie Kopierschutzanalysator und Customizer.

Bücher

Bei den Büchern befaßt sich der Autor Peter Finzel im brandneuen Assemblerbuch mit der Programmierung des Atari-Computers in Maschinensprache. Auf 169 Seiten lernen Sie hier alle Grundlagen wie Zahlensysteme, Aufbau des 6502-Prozessors, Befehlssatz und Adressierungsarten kennen. Das Assemblerbuch wird durch viele Beispiele abgerundet, die Sie direkt mit "ATMAS-II" eintippen und ausprobieren können. Doch es bleibt nicht nur bei Grundlagen. An vier längeren Programmen lernen Sie die Assemblerprogrammierung Ihres Atari-Computers an konkreten Beispielen wie PM-Grafik oder der Programmierung von Interrupts kennen.

Das Buch "Die Hexenküche" zeigt dem fortgeschrittenen Leser eine Unmenge an Tips und Tricks. Folgende Themen sind u.a. enthalten: Musikprogrammierung, ein Soundentwicklungssystem für eigene Programme, Benutzung des VBIs, Touch-Tablet, Tips zum DOS XL, neue Grafikmodi und vieles mehr. Zum Buch ist auch eine Diskette erhältlich, die alle Programme vollständig enthält.

Das Atari-Power-Buch vom Atari-Club Düsseldorf enthält viele Listings, Bauanleitungen und eine Einführung in die Grafikmöglichkeiten des Atari-Computers. Besprochen werden Bauanleitungen für ein Druckerinterface (über die Joystickports), ein Lichtgriffel und eine Erweiterung des Disklaufwerks zum Thema Kopierschutz. Daneben werden viele Tips zu ANTIC und der Display-List, zu PM-Grafik und Scrolling erläutert.

Alle Programme und Bücher können über den Verlag bestellt werden. Einen Bestellschein finden Sie auf der Seite mit der Überschrift "Take 9".

Oskar Schleimann

Eine trackreiche Angelegenheit

Tiefere Einsicht in das ST-TOS wäre oft eine gute Sache. Hier lesen Sie das nötige Know-how.

Wahrscheinlich ist es Ihnen auch schon passiert, daß Ihr wichtigstes Programm oder Ihre größte Datei versehentlich von der Diskette gelöscht wurden. Erneutes stundenlanges Eintippen war nötig. Hier ist guter Rat teuer.

16 Bit

Kann man etwas tun, um das File wiederzubekommen oder ist es unwiederbringlich verloren?

Bekanntlich wird ein Programm durch Scratchen nicht physikalisch von der Diskette gelöscht, sondern es wird lediglich der File-Eintrag im Directory so gekennzeichnet, daß es nicht mehr angezeigt wird. Dadurch ist dann auch kein Zugriff mehr möglich. Diese Kennzeichnung kann aber rückgängig gemacht werden. Wenn zwischenzeitlich kein Schreibzugriff auf die Diskette erfolgte, kann man das verlorene File restaurieren. Um solche Manipulationen (oder auch andere) vorzunehmen, ist es hilfreich, einiges über die Anordnung der Daten auf der Diskette zu wissen. Dazu soll dieser Artikel einige Hinweise geben.

Das Atari-ST-TOS ermöglicht die Benutzung verschiedener Diskettenformate. So können die Anzahl der Seiten pro Diskette, der Tracks pro Seite, der Sektoren pro Track und der Bytes pro Sektor verschieden sein. Üblich sind 80 Tracks pro Seite (Floppy SF 354 einseitig, Floppy SF 314 zweiseitig), 9 Sektoren pro Track und 512 Bytes pro Sektor. Die Tracks (Spuren) werden von 0 bis 79 (beginnend von außen) und die Sektoren von 1 bis 9 gezählt. Daraus ergeben sich 720 Sektoren entsprechend 360 KByte (zweiseitige Floppy mit 1440 Sektoren entsprechend 720 KByte). Dieser Speicherplatz steht aber nicht ausschließlich für den Anwender zur Verfügung. Zur Verwaltung der gespeicherten Daten auf der Diskette benötigt das TOS selber einige Sektoren. Dazu gehören z.B der Bootsektor, die File Allocation Table und das Directory.

Der Bootsektor befindet sich auf jeder Diskette, und zwar auf Track 0, Sektor 1. Er wird nach dem Einschalten, nach jedem Diskettenwechsel und nach jedem Reset abgefragt und enthält Informationen über das Format der Diskette und ein eventuell zu ladendes Betriebssystem (siehe Bild 1).

Die Daten des Bootsektors haben folgende Bedeutung:

Byte 0-7:
Hier stehen Informationen, die für das Booten von Betriebssystem wichtig sind. Byte 0 und 1 enthalten den Sprungbefehl, der das Einladen und Starten des Bootcodes bewirkt.

Byte 8-10:
Sie enthalten eine 24-Bit-Identifikationszahl für die Diskette. Diese ist zufällig und daher für fast alle Disketten unterschiedlich.

Byte 11-12:
Sie geben die Anzahl der Bytes pro Sektor an, in unserem Fall \$0200, also 512 Byte. (Diese und alle anderen Daten werden auf der Diskette in der Form Low-Byte/High-Byte abgelegt.)

Byte 13:
Gibt die Anzahl der Sektoren pro Cluster an. Im Normalformat sind das \$02 = 2 Sektoren.

Byte 14-15:
Sie enthalten die Zahl der für besondere Zwecke reservierten Sektoren (hier \$0001 = 1 für den Bootsektor).

Byte 16:
Enthält die Anzahl der FATs (hier \$02 = 2). Die FAT wird zweimal abgespeichert, beginnend auf Sektor 2 und Sektor 7. Letztere wird vom TOS benutzt, die erste dient als Sicherheitskopie, denn wenn die FAT nicht mehr lesbar ist, sind alle abgespeicherten Files verloren.

Byte 17-18:
Die Anzahl der möglichen Einträge im Wurzel-Directory. Hier steht in diesem Fall \$0070

> **Bootsektor enthält Informationen über Diskettenformat**

= 112. (Für das Directory sind sieben Sektoren vorgesehen. Da jeder File-Eintrag 32 Byte benötigt, ergibt sich so die Zahl von 112 möglichen Einträgen.) Die Inhalte von Ordner zählen dabei nicht mit, da sie nicht auf den Directory-Sektoren eingetragen werden.

Byte 19-20:
Hier finden wir die Anzahl der Sektoren auf der Diskette, bei einseitigen, normal formatierten Disketten also \$02D0 = 720 (80 * 9). Bei doppelseitigen steht hier \$05A0 = 1440 (80 * 9 * 2).

Byte 21:
Enthält das Formatkennzeichen für das verwendete Diskettenformat:
F8 für Single Sided / 80 Tracks
F9 für Double Sided / 80 Tracks
FC für Single Sided / 40 Tracks
FD für Double Sided / 40 Tracks

Byte 22-23:
Sie geben die Zahl der Sektoren pro FAT an (hier fünf).

Byte 24-25:
Hier ist die Anzahl der Sektoren pro Spur festgelegt (\$0009 = 9).

Byte 26-27:
Sie enthalten die Anzahl der formatierten Diskettenseiten. Bei dieser Diskette steht hier \$0001 = 1, also handelt es sich um eine einseitige.

Byte 28-29:
Sie geben die Anzahl der versteckten Sektoren an. Diese Bytes werden vom TOS nicht benutzt.

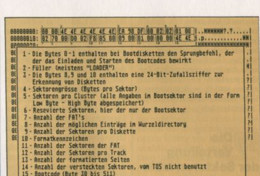
Byte 30-511:
Hier steht der Bootcode, der bei einem Reset in den Speicher geladen und gestartet wird.

Auf der Diskette befindet sich außerdem die sogenannte File Allocation Table (FAT). Sie beginnt auf Track 0, Sektor 1 und ist im Normalformat 5 Sektoren lang; bei doppelt abgespeicherter FAT werden also 10 beansprucht. In der FAT ist die Belegung der Sektoren durch die abgelegten Files gespeichert. Das ist notwendig, da die Sektoren eines Files nicht unbedingt aufeinanderfolgen müssen. Die Verteilung der Sektoren ist in Clustern organisiert. Zu einem Cluster gehören zwei Blöcke. In der abgebildeten FAT (siehe Bild 2) lauten die beiden ersten Zeilen:

F7 FF FF 03 40 0D 05 60 0D 07 80 0D 09 A0 00 0B

C0 00 0D F0 FF 0F 00 01 11 20 01 13 40 01 15 60

Die Belegung der Cluster wird mit 12-Bit-Einträgen gekenn-



```
00000000: 00 00 4E 4E 4E 4E 4E 4E EA 98 DF 00 02 02 01 00 |..NNNNNN?.?.....|
00000010: 02 70 00 D0 02 F8 05 00 09 00 01 00 00 00 4E 4E |.p............NN|
```

1 - Die Bytes 0-1 enthalten bei Bootdisketten den Sprungbefehl, der der das Einladen und Starten des Bootcodes bewirkt
2 - Füller (meistens "LOADER")
3 - Die Bytes 8,9 und 10 enthalten eine 24-Bit-Zufallsziffer zur Erkennung von Disketten
4 - Sektorengrösse (Bytes pro Sektor)
5 - Sektoren pro Cluster (alle Angaben im Bootsektor sind in der Form Low Byte - High Byte abgespeichert)
6 - Resevierte Sektoren, hier der nur der Bootsektor
7 - Anzahl der FAT's
8 - Anzahl der möglichen Einträge im Wurzeldirectory
9 - Anzahl der Sektoren pro Diskette
10 - Formatkennzeichen
11 - Anzahl der Sektoren der FAT
12 - Anzahl der Sektoren pro Track
13 - Anzahl der formatierten Seiten
14 - Anzahl der versteckten Sektoren, vom TOS nicht benutzt
15 - Bootcode (Byte 30 bis 511)

Bild 1: Der Bootsektor

File - Allocation - Table der Floppy SF 354/314

Die Sektoren sind in Clustern organisiert. Zu einem Cluster gehören zwei Sektoren (1024 Bytes). Für den Eintrag eines Clusters in die FAT werden 12 Bit, für zwei Einträge also drei Bytes benötigt. Nachstehend sind die die zusammengehörigen Bytes für je zwei Einträge eingerahmt.

```
0+1       2+3       4+5       6+7       8+9       10
F7 FF FF  03 40 00  05 60 00  07 80 00  09 A0 00  00    ?...@..`..?..?..
C0 00  0D F0 FF  0F 00 01  11 20 01  13 40 01  15 60    ?..?..... ..@..`
  +11    12+13     14+15     16+17     18+19     20+...
```

Leider kann man das mittlere Byte nicht einfach durchteilen um die Ziffern in der abgebildeten Reihenfolge zu lesen. Unten ist Vorschrift angegeben, nach der die Clustereinträge zu lesen sind, um das nächste zum File gehörende Cluster zu finden.

```
Cluster 0 und 1   Cluster 2 und 3   Cluster 4 und 5   Cluster ...

F7 FF FF          03 40 00          05 60 00

FF7   FFF         003   004         005   006         ...FFF
```

Cluster 0 ist als unlesbar, Cluster 1 als Endcluster eines Files gekennzeichnet. Cluster 2 zeigt auf Cluster 3, dieses auf Cluster 4, von dort auf Cluster 5 und von dort auf Cluster 6. Dieses Spiel geht so weiter, bis bei irgendeinem Cluster der Wert FFF als Fileendkennzeichen auftritt. Nicht belegte Cluster sind durch den Eintrag 000 gekennzeichnet.

Bild 2: Die FAT

```
00001600: 43 4F 50 59 20 20 20 20 50 52 47 10 00 00 00 00 |COPY    PRG.....|
00001610: 00 00 00 00 00 00 AE 08 B4 0C 02 00 EC 17 00 00 |......?.?.......|
00001620:                                                 |DESK5   ACC.....|
00001630: 00 00 00 00 00 00 C7 08 B4 0C 06 00 FE 16 00 00 |......?.?.......|
00001640: 44 45 53 4B 36 20 20 20 41 43 43 00 00 00 00 00 |DESK6   ACC.....|
00001650: 00 00 00 00 00 00 CA 08 B4 0C 0C 00 3C 00 00 00 |........?...<...|
```

1 Filename - 8 Bytes (kürzere Namen werden mit \$20 aufgefüllt)
2 Extension - 3 Bytes (Kennzeichnung des Filetyps)
3 Fileattribut-Byte, hier \$10 d. h. es handelt sich um einen Ordner
4 Nicht benutzter Bereich - 10 Bytes
5 Datum - 2 Bytes
6 Uhrzeit - 2 Bytes
7 Startcluster - 2 Bytes (das erste zum File gehörende Cluster)
8 Länge des Files - 4 Bytes (hier \$17EC = 6124 Bytes oder ca. 12 Sektoren)

Die Einträge sind alle in der Form Low Byte/High Byte organisiert. Fileattribut-Byte, Datums- und Uhrzeitbytes werden bitweise gelesen. Nähere Angaben dazu finden sich im Text.

Bild 3: Directory

zeichnet. Das bedeutet, daß für zwei Einträge drei Byte benötigt werden. Nachfolgend ist der Gebrauch dreier Zahlsysteme notwendig. Die Zahlen werden jeweils durch ein vorgestelltes Zeichen markiert ($ = Hexadezimalsystem und % = Binärsystem). Fehlt dieses, handelt es sich um Dezimalzahlen.

Die Verteilung der Sektoren ist in Cluster organisiert

Die ersten beiden Einträge werden die Hexbytes F7, FF und FF benutzt. Ins Binärsystem übertragen lauten die Zahlen 11110111, 11111111 und 11111111. Der erste Eintrag belegt nun die 8 Bit des ersten und die niederwertigsten vier Bit des zweiten Eintrags. Die 12-Bit-Binärzahl heißt dann 111101111111 oder hexadezimal FF7. Für den zweiten Eintrag werden dann die höherwertigen vier Bit der zweiten Hexzahl (1111) und alle acht Bit der darauf folgenden benutzt (11111111). Die 12-Bit-Binärzahl lautet hier also 1111 11111111 oder $FFF.

Mit diesen Werten haben wir gleich zwei Kennzeichnungen kennengelernt, die eine besondere Bedeutung haben. Sektoren, die nicht beschrieben werden können (meist aufgrund eines Hardwarefehlers der Diskette), erhalten eine Markierung zwischen $FF0 und $FF7, während ein Eintrag zwischen $FF8 und $FFF das letzte Cluster eines Files kennzeichnet. Einträge zwischen $003 und $FEE bedeuten, daß das Cluster zu einem File gehört. Sektoren mit dem Wert $000 sind nicht belegt und stehen zur Datenspeicherung zur Verfügung.

Wo liegen nun die Sektoren, die als nicht beschreibbar bzw. als letztes Cluster eines Files gekennzeichnet werden? Die Cluster werden fortlaufend, mit 0 beginnend, hochgezählt. Die erste Bezeichnung mit FF7 betrifft also das nullte Cluster. Es entspricht den Sektoren 15 und 16, das nächste (erste) den Blöcken 17 und 18. Diese beiden liegen in dem für Bootsektor, FAT und Directory freigehaltenen Bereich. Das erste Cluster, das zur Datenspeicherung zur Verfügung steht, besteht aus den Blökken 19 und 20 (Cluster 2). Genau dort beginnt nun das erste Programm COPY.PRG unserer Demodisk. Die Datenspeicherung fängt bei einseitigen Disketten demnach erst auf Track 2 an.

Bei doppelseitig formatierten Disketten stehen pro Track 18 Sektoren zur Verfügung. Das TOS beschreibt in einem Track zunächst die Sektoren der Seite 1, danach die der Seite 2, um dann zum nächsten Track überzugehen. Hier beginnt die Datenspeicherung demnach auf der zweiten Spur (Track 1).

Wie lauten nun die FAT-Einträge der folgenden Cluster? Die zugehörigen Hexbytes heißen:

03 40 0D 05 60 0D 07 80 0D 09 A0 00 0B C0 00 0D F0 FF 0F

Wir müßten nun die hexadezimalen Zahlen zur weiteren Verwendung in binäre umwandeln. Das ist aber leider sehr mühsam. Glücklicherweise haben die zweiziffrigen Hexadezimalzahlen eine für unseren Zweck sehr günstige Eigenschaft (zum nachfolgenden Text siehe Bild 3). Die linke Ziffer entspricht nämlich dem Wert der vier höchst- und die rechte dem der vier niederwertigsten Bits (Hi-Nibble, Low-Nibble).

Damit lassen sich die drei Byte der folgenden zwei Einträge ($03, $40, $00) wesentlich leichter lesen. Zum ersten Eintrag gehört das Byte $03 sowie die rechte Ziffer (0) des folgenden Bytes. Diese muß davor gesetzt werden; also wird daraus $003. Die linke Ziffer (4) des zweiten Bytes kommt nun hinter das dritte, womit sich $004 ergibt.

Diese Einträge bedeuten, daß das File, das mit dem zweiten Cluster (dessen Eintrag $003 lautet) beginnt, sich auf dem dritten fortsetzt. Da das File noch nicht zu Ende ist, erfahren wir aus dem Eintrag des dritten Clusters ($004), wo es weitergeht. Auf diese Weise lassen sich alle zu einem File gehörenden Cluster ausfindig machen. Das letzte Cluster enthält, wie schon gesagt, das Kennzeichen $FFF.

In unserem Beispiel ergeben sich für das auf dem zweiten Cluster beginnende File (die Adresse des ersten steht im Directory-Eintrag) die zugehörigen Cluster 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 und 13. Der Eintrag für das Cluster 13 lautet $FFF, da dies das letzte ist, das zum File gehört.

Auf die File Allocation Table folgt das Directory der Diskette. Es belegt sieben Sektoren und enthält die Informationen über die Files. Jeder File-Eintrag ist 32 Byte lang (siehe Bild 3). Im abgebildeten Directory ist der erste Eintrag das Programm COPY.PRG ($43, $4F, $50, $20, $20, $20, $20, $20). Der Name belegt acht Byte. Bei kürzeren Filenamen wird mit $20 aufgefüllt.

Jeder File-Eintrag ist 32 Byte lang

Danach folgt die drei Byte lange Extension (hier PRG: $50, $52, $47), der sich die sogenannten File-Attribute anschließen. Dafür ist ein Byte vorgesehen, das bitweise benutzt wird. Sechs Bit davon haben eine Funktion, die beiden anderen werden nicht verwendet.

Die Bits haben folgende Bedeutungen:

Bit 1 (Read only):
Wenn es gesetzt ist, kann das File nur gelesen, aber nicht geschrieben werden.

Bit 2 (Hidden):
Bit 2 und Bit 3 haben die gleiche Wirkung. Sie unterdrükken den File-Eintrag im Directory.

Bit 3 (System):
siehe oben

Bit 4 (Volume Label):
Es kennzeichnet einen Directory-Eintrag als Diskettenname.

Bit 5 (Subdirectory):
Dieses Bit ist gesetzt, wenn es sich bei dem File-Eintrag um ein Subdirectory (Ordner) handelt. Der Inhalt des Ordners ist in dem Cluster verzeichnet, der im Direktory als Fileanfang gekennzeichnet ist.

Bit 6 (Archive):
Archivstatus

Die dem Attribut-Byte folgenden 10 Byte werden vom TOS nicht genutzt. Die nächsten beiden geben die Zeit und die zwei darauffolgenden den Tag der letzten Änderung an. Die beiden Zeitbytes heißen in unserem Falle $AE und $08. Sie sind wie die zwei Datumbytes in der Form Low-Byte/High-Byte angeordnet. Die beiden Angaben werden in einer 16-Bit-Darstellung ausgewertet. Die entsprechende 16-Bit-Binärzahl heißt %0000100010101110. Die 16 Bit werden in drei Teile eingeteilt. Die höchstwertigen fünf Bit (in unserem Beispiel %00001 = 01) geben die Stunde an (0-23), die mittleren sechs Bit (für uns %000101 = 05) sind für die Minuten zuständig, und die verbleibenden fünf Bit (nun %01110 = 14) repräsentieren die Sekunden. Da mit fünf Bit nur 31 verschiedene Zahlen darstellbar sind, werden die Werte für die Sekunden mit zwei multipliziert. Die Zeit wird also jeweils im Abstand von zwei Sekunden angegeben. Die Zeit der letzten Änderung war demnach in unserem Beispiel 01:05:28 Uhr. Ihr Datum wird ähnlich aufgezeichnet. Auch hier bilden die beiden Bytes eine 16-Bit-Zahl. Die beiden Hexbytes $B4 und $0C ergeben %0000110010110100. Die höchsten sieben Bit übernehmen die Darstellung des Jahres (%0000110 = 6, zu diesem Wert muß 1980 addiert werden, um zu der richtigen Jahreszahl zu kommen), die nächsten vier Bit den Monat (%0101 = 5), die letzten fünf werden zur Aufzeichnung des Tages verwendet (%10100 = 20). Daraus ergibt sich als Termin der letzten Änderung für unser Programm der 20.05.1986.

Nach dem Datumseintrag folgt die Angabe des Filebeginns auf der Diskette (zwei Byte, hier $02, $00), die auch in der Form Low-Byte/High-Byte gehalten ist. Sie bezeichnet das erste zum File gehörende Cluster. Das File COPY.PRG beginnt mit dem Cluster 2 ($0002). Die letzten vier Bytes endlich geben die Länge des Files in Byte an. In unserem Fall stehen dort die Bytes $EC und $17. Da das ebenfalls in der Form Low-Byte/High-Byte geschieht, heißt die Zahl $17EC oder in dezimaler Schreibweise 6124 (1 * 4096 + 7 * 256 + 14 * 16 + 12). Das File ist also 6124 Byte lang. Damit ist der File-Eintrag beendet.

Noch etwas über die Behandlung von Ordnern im Directory. Bekanntlich ermöglicht das TOS die Verwendung hierarchischer Directories. Ein Ordner wird im Directory-Eintrag durch Setzen des fünften Bit (der Eintrag lautet dann $10, sofern kein anderes Bit mehr gesetzt ist) im Attribut-Byte gekennzeichnet. Der Inhalt eines solchen Ordners steht nicht im Directory auf den Sektoren 12-18, sondern in dem Cluster, das im Directory-Eintrag als das erste des Ordnerfiles gekennzeichnet ist. Die File-Einträge des Ordners sind genauso aufgebaut wie die oben beschriebenen.

Nun aber zu unserem Ausgangspunkt zurück. Diese Directory-Einträge bleiben beim Löschen eines Programms nahezu vollständig erhalten. Das TOS ändert dabei lediglich das erste Byte des zu löschenden Eintrages in den Wert $E5 = !229. Mit einem Diskettenmonitor kann man sich nun die Einträge der gelöschten Files ansehen und $E5 durch den richtigen Wert ersetzen. Wenn dieser nicht mehr bekannt ist, kann jeder Wert < > $E5 eingetragen werden. Es stimmt dann lediglich der Filename nicht mehr. Auf jeden Fall erscheint das Programm nun wieder im Listing und kann gegebenenfalls mit dem im ST-Handbuch aufgeführten Verfahren umbenannt werden.

Leider ist damit das File noch nicht wieder vollständig restauriert. Das TOS ändert beim Löschen eines Programms nicht nur das erste Byte des Namens, sondern gibt auch sämtliche zum File gehörenden Cluster durch den Eintrag $000 in die FAT wieder frei. Die FAT muß also auch noch wiederhergestellt werden. Wenn lediglich ein File gelöscht wurde, ist das relativ einfach. Die zugehörigen Cluster lassen sich durch den Eintrag $000 leicht zwischen den belegten Sektoren ausfindig machen. Schwieriger wird es schon, wenn mehrere Files gelöscht wurden. Dann müssen ihre Sektoren mit einem Diskmonitor gesucht, gefunden und in der FAT manuell wieder verknüpft werden.

Dabei hilft der Algorithmus des TOS, der die Belegung der Cluster vornimmt. Beim Speichern eines Programms werden die Cluster nur in aufsteigender Reihenfolge belegt (solange dort noch welche frei sind) und müssen also auch nur in dieser Reihenfolge abgesucht werden (bei doppelseitigen Disketten zuerst Vorderseite, dann Rückseite, anschließend der Track). Bei Dateien, die Daten in Klarschrift enthalten, kann dieses Verfahren erfolgreich angewendet werden. Bei gelöschten Programmfiles dürfte es wesentlich schwieriger sein, die zusammengehörigen Sektoren in der richtigen Reihenfolge herauszufinden.

Herbert Holstein

Maus und Musik

Musix 32 wäre zum Klimpern zu schade.

Wie bei so vielen Computern ist es auch beim Atari ST nicht einfach, die musikalischen Fähigkeiten, die der Hersteller seinem Produkt mitgegeben hat, in schöne Melodien umzusetzen. Da sind Kanäle zu beachten, Tonhöhe und -dauer zu bestimmen, Hüllkurven anzusteuern und vieles mehr. Die meisten Hobby-Anwender beschränken sich daher auch auf vereinzelte Töne oder eher spärliche Melodien. Verwundert hört man dann bei verschiedenen kommerziellen Programmen, insbesondere bei Spielen, die tollsten Soundeffekte und gute Begleitmusik und fragt sich, wie das möglich ist.

16 Bit

Vielleicht benutzen diese Autoren das Programm "Musix 32", das ich getestet habe. Voraussetzung, um damit arbeiten zu können, ist ein beliebiger ST-Computer mit Monochrom-Monitor. Außerdem sollte der zukünftige Anwender Notenkenntnisse mitbringen. Sie sind zwar nicht unbedingt erforderlich, erleichtern die Arbeit aber sehr.

Nach dem Laden erscheinen auf dem Monitor eine Notenlinie und ein umfangreiches Symbolmenü. Gesteuert wird das Programm weitgehend über die Maus. Bevor man sich an eigene Werke macht, kann man aus zahlreichen Demostücken eine Kostprobe aussuchen und laden. Die bereits vorhandenen Stücke zeigen, wie gut Musik mit dem Computer klingen kann. Zur Übung kann man am Anfang auch ein bereits fertiges Stück manipulieren; die Möglichkeiten dazu sind vielfältig. Das Funktionsmenü bietet:

- Wahl zwischen den Kanälen A, B und C
- Verschiebung des auf dem Monitor sichtbaren Teils der Notenlinie zum Anfang oder Ende, um eine Bildschirmbreite oder um eine Note
- Manipulation der Ablaufgeschwindigkeit
- Einstellung des Taktmaßes
- Eingabeecho
- Unterdrückung nicht aktiver Tonkanäle
- Abspielen der Musik komplett oder als Ausschnitt
- Einstellung von Kammerton A. Dieser liegt bei 440 Hertz und ist als Grundlage zu betrachten. Er kann hier stufenlos höher oder tiefer gestellt werden, was sich auf das gesamte Musikstück auswirkt.
- Hüllkurvenformular zur Klangvariation
- Ausdruck eines Stücks
- Ausschneiden, Einfügen und Kopieren eines Notenlinienteils
- Laden und Abspeichern von Kompositionen
- Papierkorb zur Vernichtung schlechter Stücke

Komponiert wird ebenfalls mit der Maus. Am oberen Bildrand ist eine Leiste installiert, in der die verschiedenen Symbole für Noten-, Vor- und Pausenzeichen enthalten sind. Mit dem Cursor holt man sich das gewünschte Symbol und plaziert es in der Notenlinie. Note für Note und Zeichen für Zeichen muß man so sein Stück montieren. Zur akustischen Kontrolle kann das Symbol Echo angeklickt werden. Jede eingesetzte Note wird dann auch als Ton wiedergegeben. Das Komponieren eines längeren Stückes bedeutet somit echte Arbeit.

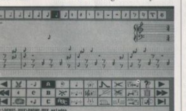
Vielleicht wäre es einfacher, die Töne direkt über die Tastatur zu spielen bzw. einzusetzen. Diese Option ist aber leider nicht vorhanden. Auch gibt es zur Zeit keine besonderen Möglichkeiten, den Klang zu verfremden oder Schlagzeug-Effekte zu verwenden. Der Hersteller von "Musix 32" arbeitet aber bereits an einem Zusatzmodul, das diese Funktionen bieten soll (Erscheinungstermin ca. Anfang 1987).

Die interessanteste Eigenschaft des Programms ist meiner Meinung nach die Möglichkeit, komponierte Stücke in eigene Programme einzubinden. Im Funkionsmenü ist dafür extra eine Option vorhanden. In der deutschen Anleitung werden darüber hinaus Beispiele gegeben, wie man unter Basic, Pascal und C die "Musix 32"-Produkte verarbeitet. Auf der Diskette sind einige Demos dazu abgespeichert.

"Musix 32" ist ein solides Programm, das den musikalischen Horizont des Anwenders erheblich erweitern kann. Sollte das erwähnte Zusatzmodul tatsächlich erscheinen und die Versprechungen erfüllen, könnte das Programm noch erheblich vielseitiger werden. Bis dahin kann man ja schon einmal ausprobieren, was bisher geboten wird.

System: Atari ST (monochrom)
Hersteller/Bezugsquelle:
Tommy Software

Stephan König



Komfortables Komponieren mit der Maus: der Bildschirm bei Musix 32

Selbst ist der Spieler

Mit unserer Serie zur Programmierung von Spielen kommen Sie der Software-Krise bei. Teil I bringt die Grundlagen für XL und XE.

Man kann immer wieder ins Staunen geraten, wenn man einen Blick über den Kanal wirft. Dort ersticken die britischen Computer-Fans förmlich in einer Flut neuer Spiele, die Monat für Monat auf den Markt geworfen werden. Diese sind teilweise von sehr hoher Qualität und stammen fast vollständig aus britischer Produktion. Bei uns dagegen sind nahezu alle angebotenen Spiele aus USA oder England importiert. Einheimische Programme sind geradezu mit der Lupe zu suchen.

Zugegeben, auch auf dem englischen Softwaremarkt sind Programme für XL/XE-Computer sehr rar, da sich Atari mit seinen 8-Bit-Maschinen in Großbritannien leider nie recht durchsetzen konnte. Aber wenn man das Angebot bei anderen Computermarken am deutschen Markt betrachtet, fällt das oben beschriebene Phänomen deutlich auf: viel Importe, wenig deutsche Produktionen.

Woran liegt das nun? Sollten wir tatsächlich ein so nüchtern denkendes Volk sein, dem zum Thema Computer nur Textverarbeitung und Tabellenkalkulation einfällt? Oder sollte es uns an Programmierern fehlen, die Ihre Fantasie in Programmcodes umsetzen können? Man sollte eigentlich weder das eine noch das andere so recht glauben. Bestimmt schlummert in so manchem Kopf eine brandneue Idee, die nur darauf wartet, in ein Programm gekleidet zu werden.

Deshalb werden Sie in dieser Serie mit Hintergrundinformationen zum Thema "Wie programmiert man ein Spiel" versorgt. Das bedeutet aber nicht, daß Sie hier fertige Spiele zum stupiden Abtippen präsentiert bekommen. Vielmehr werden Sie Anregungen und Hilfestellungen finden, um eigene Ideen in ein Programm zu packen.

Somit ist das erste Problem schon aufgetaucht: eine gute Idee, die grundlegende Voraussetzung für ein Spiel. Etwas Originelles sollte es schon sein, denn mit einem neuen "PacMan" oder "Space-Invaders" erzeugen Sie bei einem Spiele-Fan bestenfalls ein müdes Gähnen. Lassen Sie Ihrer Fantasie ruhig freien Lauf; es gibt noch eine Menge anderer Themen, die ein Computerspiel wert wären: Rollenspiele, Adventures, Simulationen und vieles mehr.

Neben der Idee benötigt man natürlich auch etwas handwerkliche Erfahrung im Programmieren und eine Portion Durchhaltevermögen, um das Projekt durchzuziehen. Aber es macht Spaß und ist schon ein Erfolgserlebnis, wenn man seine Idee so nach und nach am Bildschirm Gestalt annehmen sieht.

Natürlich ist das einfacher gesagt als getan. Bevor man mit der Programmierung beginnt, sollte man sich zunächst überlegen, mit welcher Sprache man arbeiten und wie man Logik und Grafik des Spiels anlegen will.

Eines der ersten Probleme ist die Auswahl einer geeigneten Sprache. Natürlich hängt dies stark vom Charakter des geplanten Spiels ab. Beispielsweise ist es nahezu unmöglich, ein gutes Actionspiel voll in Basic zu schreiben, andererseits muß ein reines Text-Adventure nicht unbedingt in Assembler programmiert werden. Sehen wir uns die wichtigsten für den Atari verfügbaren Sprachen mit Ihren Eigenheiten doch einmal an.

Das eingebaute Atari-Basic hat zwar einige Befehle, die Grafik und Sound unterstützen, ist aber sonst ziemlich langsam in der Ausführung. Zum Programmieren von Spielen ist es daher nur bedingt geeignet. Bessere Ergebnisse wären sicherlich mit erweiterten Versionen wie Basic XE oder Turbo-Basic zu erzielen, nur darf man natürlich nicht vergessen, daß später jeder, der in den Genuß dieses Spiels kommen will, über die gleiche Sprache verfügen muß.

Durch das Compilieren eines Basic-Programms ist in vielen Fällen ein enormer Geschwindigkeitszuwachs zu verzeichnen, allerdings ist er noch viel zu gering,

8 Bit

um damit fließende Bewegungen zu ermöglichen. Compilierte Basic-Programme bieten sich jedoch für Text-Adventures oder Strategiespiele an.

Assembler ist sicherlich die Sprache, in der die meisten Spiele geschrieben sind. Assembler-Programme sind sehr schnell und eignen sich daher besonders für Animation und Bewegung. Andererseits ist Assembler aber sehr langwierig zu programmieren.

Action! ist die einzige Compilersprache, die (außer Assembler) zur Programmierung von Spielen mit viel Bewegung voll geeignet ist. Die Geschwindigkeit eines Action!-Programms kommt schon sehr nahe an Assembler heran. Es ist sogar möglich, Interrupt-Routinen für VBI und DLI mit dieser Sprache zu programmieren. Ein schlimmer Mangel ist allerdings, daß die damit geschriebenen Programme nur laufen, wenn die Action!-Cartridge im Rechner steckt. Glücklicherweise gibt es einige Tricks, die es erlauben, solche Programme vom Steckmodul unabhängig zu machen. Wer damit arbeiten will, sollte die Serie "Action!-Center" verfolgen, die ebenfalls in diesem Heft beginnt.

Daneben gibt es noch Pascal- und C-Compiler für Atari-Computer, die sicherlich auch zur Spieleprogrammierung einsetzbar wären. Besonders für mehr auf Strategie zielende Spiele wäre das Kyan-Pascal bestimmt nicht schlecht, da es den Komfort einer strukturierten Sprache mit der Möglichkeit verbindet, zeitkritische Programmteile in Assembler nahtlos einzufügen. C-Compiler für den Atari gibt es

"Basic ist für Animation ungeeignet"

zwar, sie sind aber in Deutschland so gut wie nicht erhältlich. Wir können diese Sprache also aus unseren Betrachtungen (vorerst) ausklammern.

Sie sehen, jede Programmiersprache hat ihre eigenen Vorzüge und Nachteile. Es ist daher keine schlechte Idee, mehrere Sprachen miteinander zu kombinieren, soweit das eben möglich ist. Eine gute Mischung ergibt z.B. Basic mit Assembler. Man schreibt alle zeitkritischen Routinen in Assembler, während die Oberaufsicht bei Basic verbleibt. Auf diese Weise kann man recht gute Animation erzeugen, ohne das Basic völlig verlassen zu müssen. Von meinem Spiel "Cavelord" gab es auch eine Version, die zu einem guten Teil in Atari-Basic geschrieben war. Während alle Routinen zum Bewegen der Players und zum Steuern des Scrollings Maschinenprogramme waren, verwaltete Basic die Punkte und den Ablauf des Spiels.

Wie man so etwas macht, werden wir uns später näher ansehen. In der nächsten Folge wollen wir uns mit den Grafikfähigkeiten des Atari auseinandersetzen und dabei sehen, wie man sie für Spiele einsetzen kann.

Peter Finzel

CAD – 3 D auf ST

Nicht alles, was sich hinter solchen Kürzeln verbirgt, lohnt eine nähere Betrachtung. Dieses Programm aber auf jeden Fall.

Für einen Computer mit Grafikfähigkeiten, wie sie die Atari-Computer aufweisen, war es abzusehen, daß über kurz oder lang eine Menge Programme auftauchen würden, die das Kürzel "CAD" in Programm oder Beschreibung verwenden. Aber was bedeutet eigentlich CAD? Wie viele Computer-Fachwörter ist auch CAD ein Kunstwort aus den Anfangsbuchstaben mehrerer englischer Begriffe: CAD steht für Computer Aided Design, was soviel wie "computergestützter Entwurf" bedeutet. Streng genommen ist also jedes Zeichenprogramm ein kleines CAD-System. In der Industrie jedoch versteht man unter CAD-Systemen Computer, auf denen Maschinen oder andere Güter (z.B. Autos, Mikroprozessoren, Verpackungen) entworfen werden. Oft geschieht dies in Verbindung mit rechnergesteuerten Werkzeugmaschinen, so daß man von CAM (Computer Aided Manufactoring) spricht: Der computergestützten Fertigung.

16 Bit

Am besten erkennt man die Fähigkeiten dieses Programmes aus der abgedruckten Hardcopyserie: Bild 1 zeigt die 4 Fenster von CAD-3D zusammen mit der Info-Box des Autors. Um Körper einzugeben, gibt es unter dem Menüpunkt Generate zwei Möglichkeiten: Spin und Extrude. Mit Spin erzeugt der Computer aus dem Umriß einer Figur einen dreidimensionalen Körper, indem er diesen um den Mittelpunkt dreht. Es muß dazu nur eine Hälfte eingegeben werden, die andere wird durch Spiegelung an der vertikalen Achse erzeugt.

In Abbildung 2 erkennt man den Umriß eines Kelches, den man mit der Funktion "Do Spin" berechnen lassen kann. Bild 3 zeigt das Ergebnis, wobei aus dem Modes-Menü der Punkt "Hidden" (verdeckte Linien) verwendet wurde: Alle verdeckten Linien werden nicht gezeichnet. In den 3 anderen Fenstern sieht man den Kelch von oben, rechts und von vorn.

In der Mitte des Schirmes wird in diesem Bild auch die zugehörige Infobox dargestellt, was einen Eindruck von der Speicherintensität solcher Programme vermittelt. Es müssen schließlich pro Punkt 3 Koordinaten abgespeichert werden. Hinzu kommen die Linientabellen, die angeben, welcher Punkt mit welchem verbunden ist und welche Linien eine Fläche bilden. Hier wird klar, warum 8-Bit-Computer mit solchen Programmen überfordert sind.

Bild 4 zeigt den Kelch, nachdem die Camera auf das volle Bildschirmformat vergrößert wurde. Gleichzeitig wurde die Option "Outlined" gewählt, damit der Körper entsprechend schattiert wird. Schaltet man die Fensterfunktionen auf "Rotate" um, so kann man durch Bewegen der Schieber des Camera-Fensters auch schräg in den Kelch blicken. Bei dieser Abbildung wird eine Lichtquelle simuliert, die sich in der Mitte über der Camera befindet. Bis zu 3 Lichtquellen, deren Position individuell bestimmt werden kann, sind gleichzeitig möglich.

Bild 5 gehört zu den Demo-Dateien, die sich auf der Programmdiskette befinden: Ein Wasserhahn in hervorragender dreidimensionaler Darstellung.

Bild 6 demonstriert die zweite Generate-Funktion: Extrude. Hier werden Körper nicht durch Drehung erzeugt, sondern aus dem Grundriss "in die Höhe gezogen". Auch dieses Bild befindet sich bereits auf der Diskette.

Das CK-Logo hingegen entstand bei eigenen Versuchen mit der Extrude-Funktion (Bild 7).

Bild 8 stammt wieder von der Programmdisk: die komplette Stonehenge-Anlage, die man nun beliebig betrachten kann.

An der ausgeklappten File-Box fällt die Funktion SAVE.PIC auf: Damit kann eine Ansicht als Bild abgespeichert werden, wobei die wichtigsten Formate zur Verfügung stehen: Degas und Neochrome. Allerdings kann bei Monochrom nur ersteres gewählt werden. Eine Anwendungsmöglichkeit hierfür wären zum Beispiel 3-D-Schriftzüge, die man dann mit Degas oder Neochrome ausarbeiten und als Titelbild o.ä. verwenden kann. Auch der scheinbar einfache Kelch, der sehr einfach zu konstruieren ist, macht sich als fein schattiertes Farbbild hervorragend!

Damit wären wir beim Farbmonitor: CAD-3D arbeitet sowohl mit Farbe als auch mit S/W-Monitor. Allerdings muß bei Farbe der vierfarbige Mid-Res-Mode eingestellt werden. Auch mit Farbmonitor ändert sich das Erscheinungsbild des Programmes kaum. Farbe kommt erst mit der Option SUPER-VIEW ins Bild, was der Darstellung entspricht, wie sie dann auch als Bild ausgegeben wird.

Auch Animation kann man vom File-Menü aus anwählen: Veränderungen in der Position können mitgeschnitten und abgespeichert werden. Mit dem Programm ANIMATE lassen sich solche Bewegungsabläufe dann darstellen. Das dritte Programm auf der Diskette ist SLIDEANI, mit dem sich mehrere Animationssequenzen als Endlosshow darstellen lassen (ähnlich wie das bekannte Neochrome-Demo).

Obwohl mit CAD-3D bereits sehr schnell tolle Grafiken erstellt werden können, erfordert es indessen einige Übung, bis man Files erzeugen kann, die den beigelegten Demos nahekommen. Da man jedoch auch mehrere Körper zu Gruppen zusammenfassen kann, ist es möglich, Bibliotheken mit gelungenen Körpern anzulegen und immer wieder zu verwenden.

System: Atari ab 1 MByte, s/w oder Farbe

Preis: ca. 199.– DM

Thomas Tausend

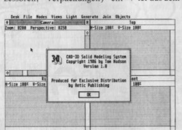


Bild 1

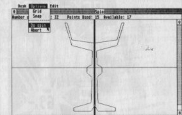


Bild 2

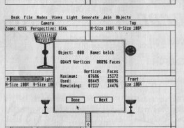


Bild 3

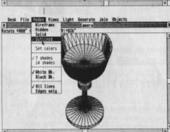


Bild 4

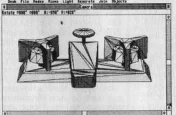


Bild 5



Bild 6



Bild 7

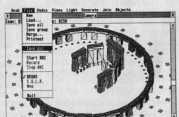


Bild 8

Nicht meisterlich

Print Master kennt man bereits von anderen Computern. Die Version für den ST nutzt dessen Möglichkeiten jedoch nicht aus!

Wer sich auf dem Softwaremarkt – gleich für welchen Computer – schon einmal nach Programmen für seinen Drucker umgesehen hat, wird sicher auch schon auf "Print Master" oder das bekanntere Vorbild "Print Shop" gestoßen sein. Beide Programme dienen zur einfachen, menügesteuerten Gestaltung von Plakaten, Grußkarten, Briefköpfen, Bannern und (bei "Print Master") auch von Monats- und Wochenkalendern.

16 Bit

Das erste Menü, das nach dem Start von "Print Master" erscheint, ermöglicht die Auswahl zwischen den oben genannten Funktionen. Zusätzlich können auch der Grafik-Editor, die Druckeranpassung und eine Quit-Funktion (Go to Desktop) angewählt werden. Arbeitet man zum ersten Mal mit diesem Programm, so sollte man zuerst die Druckeranpassung verwenden, um es auf den verwendeten Printer einzustellen. Für alle gebräuchlichen Modelle (von Bluechip über Epson und IBM bis Star und Toshiba) sind Treiber vorhanden. Da die meisten Drucker Epson-kompatible Steuerzeichen verwenden, gibt es nur wenige, die mit "Print Master" nicht benutzt werden können.

Die Auswahl der dargestellten Funktionen geschieht im ganzen Programm mit den Cursor-Tasten und RETURN oder mit der Maus, jedoch nicht durch das gewohnte Anklicken. Der Cursor bewegt sich in Bewegungsrichtung der Maus mit, so daß für die sichere Auswahl eines Punktes etwas Übung notwendig wird. Sollten Sie trotzdem einmal eine Auswahl irrtümlich getroffen haben, können Sie diese jederzeit mit ESC rückgängig machen. So läßt sich vom Ende des Gestaltungsvorgangs bis zum Anfang zurückblättern, um dort etwas zu verändern. Arbeitet man dann normal weiter, so werden die bereits vorher getroffenen Punkte als Voreinstellung invertiert, so daß man nur noch RETURN zu drücken braucht.

Anhand eines Plakates möchte ich nun die Entstehung eines solchen Kunstwerkes erklären. Nachdem man sich also für SIGN entschieden hat, verlangt "Print Master" als erstes die Quelle, aus der die Umrandung gelesen werden soll. Default ist hier die Systemdiskette, auf der einige Ränder gespeichert sind. Finden Sie hier nicht die richtigen, so können sie auch von anderen Disketten geladen werden.

Jetzt dürfen Sie bereits die erste Grafik wählen; auf der Systemdiskette sind einige Dutzend gespeichert. Die Shapes müssen hierbei nicht nach den (englischen) Namen alleine ausgesucht werden, sie können auch in einem Fenster am rechten Bildschirmrand durchgeblättert werden. Hat man seine Wahl getroffen, so fragt "Print Master" nach der Größe. Drei Stufen stehen dem Anwender dabei zur Auswahl. Die Grafik kann nun, je nach gewählter Größe, an verschiedenen Stellen eingesetzt werden. Leider sind hier nur einige Positionen fest vorgeschrieben. Auf einem Computer wie dem Atari ST wäre es sicher möglich gewesen, die Grafiken auch stufenlos zu verschieben. Selbstverständlich kann dieselbe Grafik auch mehrmals auf einer Seite vorkommen, ja man kann sogar viele kleine Bildchen nahtlos oder ziegelartig versetzt verwenden.

Die gleichen Funktionen werden nun auch für die zweite Grafik verwendet. "Print Master" kann auf einer Seite maximal zwei (verschiedene) Shapes drucken, selbstverständlich in verschiedenen Größen. Leider hat keine der beiden Grafiken Vorrang, so daß die Bilder manchmal übereinander gedruckt werden, was nicht gut aussieht.

Dem Kunstwerk kann auch Text hinzugefügt werden. Genau wie bei Rand und Grafiken erkundigt sich das Programm auch hier, ob eine andere Quelldiskette verwendet werden soll. Die Menge des Textes, der eingegeben wird, hängt vom gewählten Zeichensatz ab. Er kann links- oder rechtsbündig geschrieben bzw. zentriert werden. Für jede Zeile sind ein anderer Zeichensatz, normale oder doppelte Buchstabengröße und verschiedene Styles (normal, OUTLINED oder 3-D) möglich.

Im letzten Menü kann man jetzt seine Kreation abspeichern oder ausdrucken. Bevor der Drucker jedoch wirklich an die Arbeit geht, wird das Blatt auf dem Bildschirm dargestellt. Sieht man nun, daß sich z.B. Text und Grafik oder Grafiken untereinander unschön überlappen, so fährt man mit ESC bis zur entsprechenden Stelle zurück und nimmt die entsprechenden Änderungen vor. Leider erhält man diese Möglichkeit erst am Schluß, nachdem alle Eingaben gemacht wurden. Dies ist aber immer noch besser, als wenn man

erst nach dem Drucken (was dann ja doch einige Zeit dauert) seine Fehler bemerkt. Ein letztes RETURN bringt das Kunstwerk zu Papier.

Bei Grußkarten wird der eben beschriebene Vorgang zweimal durchlaufen, einmal für die Umschlags- und einmal für die Innenseite. Während das Plakat ca. DIN A4 mißt, ist eine Grußkartenseite nur ein Viertel so groß. Beim Drucken fällt auf, daß die Innenseite oben links gedruckt wird, und zwar auf dem Kopf! Die Außenseite wird dann unten rechts plaziert. Nachdem man das Ergebnis jedoch zweimal gefaltet hat, ist alles am rechten Fleck, und die Karte ist fertig. Mit etwas Phantasie kann man so einfach (und billig!) originelle und individuelle Geburtstags-, Weihnachts- und sonstige Glückwunschkarten herstellen, die nicht in der Flut der kommerziellen Glanz- und Kitschwerke untergehen. Wenn man sich beim Papierladen um die Ecke ein paar Bogen farbiges Papier besorgt, ist die Wirkung garantiert. Übrigens liegt auch der "Print Master"-Packung eine Bestellkarte für buntes Papier bei, wobei allerdings das Porto (Ausland!) teurer kommen dürfte als das Papier.

Die Banner-Option wird sicher selten verwendet. Mit ihr lassen sich lange Spruchbänder herstellen. Der Text wird dabei längs gedruckt und reicht über die ganze Papierbreite.

Nützlicher erscheinen mir die Briefkopf- und Kalenderfunktionen, bei denen die Grafiken jeweils ähnlich aufgebaut werden (s. Beispiele). Beim Monatskalender ist zu erwähnen, daß das Programm die Wochentage selbst berechnet. Der Anwender braucht lediglich Jahr und Monat einzugeben.

Wer mit den vorgegebenen Bildern nicht auskommt, kann vom Hauptmenü aus den Grafik-Editor aufrufen. Mit ihm können eigene Bilder erstellt oder vorhandene verändert werden. Wer "Print Shop" z.B. vom Atari XL oder Apple kennt, wird den zusätzlichen Komfort zu schätzen wissen.

"Print Master" ist ein nützliches Programm, das zwar den verschiedenen Versionen auf den 8-Bit-Computern einiges voraus hat, die Möglichkeiten des ST jedoch kaum nutzt. Hier hätte man einiges besser machen können.

System: Atari ST s/w + Drucker
Hersteller: Unison
Preis: ca. 198.– DM

Thomas Tausend

Wenn Sie lieber in der Nase bohren als seitenlange Listings abzutippen, dann sollten auch Sie von unserem Software-Service **LazyFinger** *Gebrauch machen.*



- *Gem-Routinen*
- *Dateiinfo*
- *Puzzler*
- *Nützliche Routinen für Assemblerprogrammierer*
- *Zusatz: 3D Flying Ace mit SW-Monitor (aus CK 11/86)*

Best.-Nr. LF 16-187

- *XL-TOS*
- *Kreisler*
- *Vectorgrafik in Action*
- *Listing zur Happy-1050-Diskettenstation*

Best.-Nr. LF 8-187

16 Bit **8 Bit**

Jede Diskette kostet DM 20,-. *Bitte bestellen Sie bei ATARI magazin, Postfach 1640, 7518 Bretten. Bei Versand per Nachnahme werden Versandkosten von DM 5,70 berechnet, bei Vorauskasse (Scheck beilegen oder überweisen auf Postscheckkonto Karlsruhe 43423-756) werden keine zusätzlichen Gebühren fällig.*

Aus unseren Computer Kontakt -Heften können Sie die folgenden Programm-Disketten bestellen:

A 10 *Lunar Lander (12/84), Car Race (7/84), Turbo Worm (1/85), Munsterjagd (3/85), Bewegte Grafik (3/85), Digger (2/85), 15 und 3 (4/85), Bundesligasimulation (3/85), *3-D Laby (10/84), Zeichensatzeditor (2/85), Mini-Trickfilmstudio (9/85), Rolly Dolly (11/84), *Musik-Editor (4/85). *Programme nur mit Erweiterung lauffähig.*

A 11 *Sound Demo I (5/85), Sound Demo II (nicht veröffentlicht), The Run and Jump Construction Set (6/85), *Bank Panik (7/85), Funktions-Plotter (5/85), Blockade (9/85), Jewel Eater (5/85), Zeilen-Assembler (7/85), Joystick-Controller (9/85), Horizontales Scrolling (5/85), Converter [DOS III in DOS II] (9/85). *Programme sind nur mit Erweiterung lauffähig.*

A 12 *DL Designer 64K (10/85), Joypaint 64K (10/85), Musicreator 64K (11/85), Chefredakteur 64K (1/85), Unprotector V 1.0 16K (1/85), Key Maker 16K (1/85)*

A 13 *Cherry Harry (3/86), Mission X auf dem Atari (5/86), Basic-Erweiterung (5/86), Mini-Billard (10/85), Zeichen-Zauberer (3/86), Sound-Demo (3/86).*

A 14 *Revolver Kid (1/86), Fys-Dos (7/86), Text im Grafikfenster (7/86), Rollerball (7/86), Kung Fu (9/86), Disk Menü (9/86), Titan (9/86).*

A 15 *Der hungrige Goff (11/86), Atari-Puzzler (11/86), Karteiverwaltung (11/86), Disc-Collector (11/86), Midi-Disk-Programm (11/86), MicroMon [nur für Kassettenbetrieb] (11/86), Wombel (1/87), Calc 800 (1/87), Diskeditor (1/87), Speed Tape (1/87), Filecopy (1/87), Zeichensatzfinder (1/87), Hardcopy GP 500 AT (1/87).*



Die Zahlen in Klammern bezeichnen die Ausgabe der Zeitschrift **Computer Kontakt**, *in der das Programm erschienen ist. Sie erhalten alle Programme inklusive Anleitung für nur DM 20,- pro Disk. Bitte bestellen Sie beim ATARI magazin, Postfach 1640, 7518 Bretten. Bei Zahlung per Nachnahme werden Versandkosten von DM 5,70 berechnet, bei Vorauskasse (Scheck beilegen) sind keine zusätzlichen Kosten zu entrichten.*

Blitzschnelle Vektoren

Dieses erste Action!-Center befaßt sich mit Grafikanimation auf den 8-Bit-Ataris.

Obwohl die Softwareflut bei den 8-Bit-Ataris längst nicht so hohe Wellen wie bei Schneider-Computern oder dem C 64 schlägt, gibt es doch einige "Software-Perlen", um die uns die Besitzer der obigen Computertypen mehr als beneiden. Eine davon ist die Programmiersprache Action!. Dies ist eine strukturierte Compilersprache nach Art von Pascal bzw. C, die zu purer 6502-Maschinensprache übersetzt wird und damit Laufzeiten erreicht, die sonst nur Assembler-Programmen vorbehalten sind.

Dabei ist Action! so einfach zu programmieren wie Basic, nur eben ein wenig anders, da es sich um eine strukturierte Sprache handelt. Mit anderen Worten, es gibt kein GOTO, dafür aber eine Reihe von Verzweigungs- und Schleifenstrukturen, wodurch der Sprungbefehl mehr als überflüssig wird. Lassen Sie sich durch das Fehlen von Zeilennummern und dem Einrücken der Zeilen nicht abschrecken. Letzteres ist nur ein Stilmittel zur Hervorhebung der Struktur. Natürlich könnte man Action!-Programme auch in bester Basic-Manier schreiben (möglichst viel in eine Zeile). Doch bringt dies nichts; das compilierte Programm wird weder kürzer noch schneller.

Wo soviel Licht ist, darf natürlich ein wenig Schatten auch nicht fehlen. Sicherlich der negativste Aspekt an Action! ist der hohe Anschaffungspreis, der immerhin in der Größenordnung eines neuen 800 XL liegt. Die Programme sollen nur mit eingestecktem Action!-Modul laufen, doch glücklicherweise haben Handbücher nicht immer recht, und es gibt eine Anzahl von Tricks, mit denen man Action!-Programme vom Modul unabhängig macht. Wer meine Assemblerecke in der Zeitschrift CK-Computer Kontakt verfolgt hat, weiß bereits Bescheid.

Genug der Vorrede, es wird höchste Zeit, daß wir uns mit einem konkreten Beispiel beschäftigen. Wie Sie nun wissen, ist Action! enorm schnell und daher genau die richtige Sprache zum Programmieren von flinker Grafik. Nein, wir wollen nicht schon wieder die Player-Missiles strapazieren, sondern diesmal Animation in hochauflösender Grafik produzieren. Genauer ausgedrückt wird es sich um Vektorgrafik handeln. Was versteht man nun darunter?

8 Bit

In normaler (Raster-) Grafik ist ein Objekt (Shape, Sprite, Bob usw.) aus einzelnen Punkten aufgebaut. Bei Vektorgrafik dagegen wird ausschließlich mit Linien (Vektoren) gezeichnet. Zur Definition eines Objekts gibt man die Anfangs- und Endpunkte aller dazu benötigten Linien an. Der enorme Vorteil liegt darin, daß Vektorobjekte praktisch stufenlos vergrößert und verdreht werden können, indem man die Koordinaten der Anfangs- und Endpunkte mathematisch umrechnet. Im Beispiel werden wir uns ein Programm näher ansehen, das solche Objekte zeichnen und vergrößern kann.

Wer ab und zu in eine Spielhalle geht, kennt sicherlich "Star Wars" oder das beinahe schon historische "Asteroids"; beide arbeiten mit reiner Vektorgrafik. Die Automaten verfügen über spezielle Hardware (Vektor-Displays), die nur auf diese Grafik ausgelegt sind.

So etwas steht uns im Atari zwar nicht zur Verfügung, doch können wir es leicht nachahmen. Man benötigt dazu nur hochauflösende Grafik und ein Programm, das sehr schnell Linien zeichnen kann. Beides ist für den Atari kein Problem, denn schließlich besitzt er den ANTIC-Chip für Grafik und Action! für flotte Programme.

Wir dürfen allerdings nicht den Action!-Befehl DRAWTO zum Linienzeichnen verwenden. Er arbeitet nämlich mit einer Routine des Betriebssystems, die auch von Basic benutzt wird und nicht gerade schnell ist. Wesentlich besser ist es, die DRAWTO-Routine selbst in Action! zu schreiben (im Programm ist das die Routine LineTo). Es ist kaum zu glauben, aber sie ist tatsächlich schneller als ihr in Assembler programmiertes Gegenstück im ROM des Betriebssystems.

Jetzt zu den Vektorobjekten. Ein solches wird im Programm in einem Byte-Array abgelegt. Das erste von je drei aufeinanderfolgenden Bytes bestimmt, ob ein neuer Anfangspunkt festgelegt oder eine Linie gezogen werden soll. Das kann man sich bildlich so vorstellen, daß das Objekt auf kariertes Papier gezeichnet würde. Bei einer Null wird der Stift abgehoben und nur der neue Punkt angesteuert; eine Eins hingegen bedeutet, daß eine Linie vom letzten zum neuen Punkt gezogen wird. Ein Wert von $FF zeigt an, daß das Objekt fertig gezeichnet ist. Das zweite und dritte Byte geben die Koordinaten des neuen Punktes (zuerst X, dann Y) an.

Im Programm wurde als Beispiel das Atari-Logo (im Array Atari_L) abgelegt. Zum Entwurf eines Objekts zeichnet man es auf kariertes Papier und überträgt die Koordinaten nach der oben geschilderten Methode in ein Byte-Array. Beim Entwurf sollte man sich an eine Größe von 10 × 12 Kästchen (Höhe × Breite) halten, da sonst die Mitte des Objekts nicht richtig berechnet wird (oder die DEFINEs Mitte_X und Mitte_Y ändern).

Die Prozedur Draw( ) zeichnet ein gesamtes Objekt in einen HiRes-Bildschirm. Sie können dabei noch im Parameter VERGR angeben, um wieviel das Objekt vergrößert oder verkleinert werden soll. Ein Wert von 0 bis 9 verkleinert, 10 bildet es in Originalgröße ab, während es bei höheren Werten vergrößert wird. Daneben können noch Werte zur Verschiebung in horizontaler und vertikaler Richtung angegeben werden.

Animation kann mit der Routine Draw( ) erzeugt werden, indem man das Objekt zeichnet, dann den Bildschirm löscht und es an einem anderen Platz oder mit anderer Vergrößerung neu zeichnet. Der häßliche Nachteil dieser Methode ist nur, daß durch das dauernde Löschen und Neuzeichnen ein unruhiges und flimmerndes Bild entsteht, wodurch die Animation kaum mehr als solche erkennbar ist.

Daher wurde im Programmbeispiel zu einer List gegriffen. Man verwendet zwei Bildschirme, von denen einer zu sehen ist, während der andere gelöscht und neu gezeichnet wird. Die Umschaltung der beiden Bildschirme erledigt die Prozedur Switch_Screen( ). Durch diese Technik wird die Animation fließend.

Das Hauptprogramm nützt die Möglichkeiten der Draw( )-Routine, um zwei Atari-Logos nebeneinander abwechselnd zu verkleinern und zu vergrößern. Der Effekt ist recht plastisch; je eines der beiden Fuji-Symbole scheint in der Tiefe des Raumes zu verschwinden, um danach wieder neu aufzutauchen.

Das Programm verwendet einige Tricks und Kniffe, die Sie auch in eigenen Programmen gewinnbringend einsetzen können. Da wäre zunächst die Definition einiger Variablen in der Zero-Page. Mit zwei SET-Anweisungen wird Action! instruiert, die Definition von row bis hin zu yf in der Zero-Page ab der Speicherzelle $F0 vorzunehmen. So kann Action! diese Variable (vor allem den Array-Zeiger row) wesentlich schneller ansprechen. Nach ihrer Definition wird die Adresse des Objectcodes durch zwei weitere SET-Befehle auf die Adresse $7000 verlegt.

Mit den SET-Anweisungen kann man die Code-Erzeugung wie bei einem Assembler-Programm mit ORG (bzw. mit "*"=) steuern. Normalerweise würde das von Action! erzeugte Objektfile direkt nach dem Textfile abgelegt. Diese Methode würde sich beim vorliegenden Programm jedoch nicht anbieten, da eine Display-List definiert wird, die durch gewisse Einschränkungen des ANTIC-Bausteins keine 1 KByte-Grenze überschreiten darf. Legt man jedoch die Anfangsadresse fest auf $7000 und definiert die Display-List am Anfang des Programms, wird dieses Problem vermieden.

Die Display-List wird einfach durch mehrere BYTE- und CARD-Definitonen erzeugt. Verwendet wird ein Modus, der GRAPHICS 6 entspricht (Auflösung 160 mal 96, zwei Farben). Natürlich könnte man in diesem Falle auch einen GRAPHICS-6-Aufruf benutzen, aber da wir später mit Page-Flipping arbeiten wollen, ist die erste Methode eleganter. Es genügt dann, der Variablen LMS einen anderen Wert zuzuweisen, um die Adresse des Videospeichers zu verändern.

Anschließend folgen einige Byte-Arrays, die das Objekt sowie eine Adreßtabelle enthalten. Letztere ist für die Fast_Plot( )-Routine nötig, damit die Adressen der Zeilenanfänge möglichst schnell herausgefunden werden können. Auch hier ein Trick. Anstatt ein großes CARD-Feld zu verwenden, werden LSBs und MSBs der Adressen in getrennten Byte-Feldern aufbewahrt. Auf diese Weise kann der Zugriff viel schneller erfolgen.

Nun folgt die Fast_Plot( )-Routine, in der auch einige wirkungsvolle Tricks versteckt sind. Schreibt man nämlich zwischen Namen und Parameterklammer ein '=*', so verzichtet Action! darauf, die Parameter in lokale Variablen abzulegen. Man kann das dann selbst mit einem kleinen Codeblock erledigen. Die beiden Parameter werden aus dem X- und Y-Register in einen reservierten Zero-Page-Speicherbereich von Action! gebracht. Da es nun nicht mehr weiß, wo die beiden zu finden sind, legt man zwei adressierte Variablen (im vorliegenden Fall X und Y) darauf. Der indirekte Zugriff auf den Videospeicher geschieht über den Array-Zeiger row, der mittels Adreßtabellen aus LSB und MSB zusammengesetzt wird.

Die LineTo( )-Routine stammt aus dem Programm "View 3D" von Paul Chabot (Antic 6/85). Sie ist sehr schnell, da nur Additionen und Subtraktionen und die Fast_Plot( )-Routine benutzt werden. Die Prozedur Graphic_Init( ) aktiviert die Display-List und bereitet Adreßtabelle und das Video-RAM vor, während Screen_Switch( ) zwischen den beiden Bildschirmen (die übrigens bei Adresse $8000 bzw. $8800 beginnen) hin- und herschalten kann.

Damit wären wir am Ende des ersten Action!-Centers angelangt. Ich hoffe, es hat Ihnen gefallen und Sie haben einige neue Anregungen bekommen. Im nächsten Heft werden wir besprechen, wie man Interrupts in Action! programmieren kann. Ich würde mich freuen, wenn Sie wieder mit von der Partie sind.

Peter Finzel

Vektorgrafik in Action!

```
;************************************
;      VEKTORGRAPHIK IN ACTION!
;
;P. FINZEL                      1986
;************************************

DEFINE VRAM1  ="$8000", ;Screen 1
       VRAM2  ="$8800", ;Screen 2
       VRLen  ="1920",  ;Laenge Screen
       VMax   ="40",    ;max. Vergroesserung
       Mitte_X="6",     ;Mitte des
       Mitte_Y="5",     ;Objektes
       MODE   ="$B$B$B$B$B"
;
;Zero-Page Variable
;
SET $E=$F0 SET $F=0
BYTE ARRAY row
BYTE rowl=row, rowh=row+1
BYTE t,a,b,Xnow,Ynow
BYTE dx,dy,xf,yf
;
;Programm ab $7000 ablegen
;========================
;
SET $E  =$7000
SET $491=$7000
;
;
;Variablen und Daten
;===================
;
CARD dlist =560 ;Display-List Zeiger
BYTE color0=708 ;Schattenreg. Farbe 1
;
CARD Wrk  =[ $8000 ] ;Zeiger auf bearbeiteten
BYTE Wrkh =Wrk+1     ;Screen (Wrkh ist MSB)
BYTE scr  =[0]       ;momentaner Screen

;------------------------------------
; Die Display-List:
;------------------------------------
BYTE DLST0=[ $70 $70 $70 $4B ]
CARD LMS  =[ $8000 ]
BYTE DLST1=[ MODE MODE MODE MODE
             MODE MODE MODE MODE
             MODE MODE MODE MODE
             MODE MODE MODE MODE
             MODE MODE MODE $41 ]
CARD DJMP =[ 0 ]

;------------------------------------
; Adresstabelle
;------------------------------------
BYTE ARRAY
  adrl(96),adrh(96),
  mask8(0)=[128 64 32 16 8 4 2 1]

;------------------------------------
;Objekt im 10x12 Raster:
;------------------------------------
BYTE ARRAY ATARI_L=[
0  3  0:1  4  0:1 4 5:1 1 10:1 0 10
1  3  5:1  3  0:0 5 0:1 7  0:1 7 10
1  5 10:1  5  0:0 8 0:1 9  0:1 9  5
1 12 10:1 11 10:1 8 5:1 8  0:$FF$0$0]
;
;------------------------------------
;Graphikpunkt setzen
;------------------------------------
PROC Fast_Plot=*(BYTE x1,y1)
BYTE X=$A0, Y=$A1
BYTE xb=$A2, Xr=$A3

[ $85 $A0 $86 $A1 ]
IF Y<96 THEN
      rowl=adrl(y) rowh=adrh(y)+wrkh
      xb=x RSH 3:xr=x AND 7
      row(xb)== % mask8(xr)
FI
RETURN

;------------------------------------
;Graphik-Linie ziehen
;------------------------------------
PROC LineTo(BYTE x,y)
BYTE i
Fast_Plot(xnow,ynow)
IF x=xnow AND y=ynow THEN RETURN FI
IF x>xnow THEN
     dx=x-xnow:xf=1
ELSE
     dx=xnow-x:xf=$FF
FI
IF y>ynow THEN
     dy=y-ynow:yf=1
ELSE
     dy=ynow-y:yf=$FF
FI
x=xnow:y=ynow
IF dx>dy THEN
  a=dy+dy:t=a-dx:b=t-dx
  FOR i=1 TO dx
    DO
    x==+xf
    IF t>127 THEN
       t==+a
    ELSE
       t==+b:y==+yf
    FI
    Fast_Plot(x,y)
  OD
ELSE
  a=dx+dx:t=a-dy:b=t-dy
  FOR i=1 TO dy
    DO
    y==+yf
    IF t>127 THEN
       t==+a
    ELSE
       t==+b:x==+xf
    FI
    Fast_Plot(x,y)
  OD
FI
xnow=x:ynow=y
RETURN

;------------------------------------
;Page-Flipping: Screen wechseln
;------------------------------------
PROC Switch_Screen=*()
IF scr=0 THEN
       Lms=Vram2
       Wrk=Vram1
ELSE
       Lms=Vram1
       Wrk=Vram2
FI
scr==+1&1
Zero(Wrk,VRLen)
RETURN

;------------------------------------
;Display-List & Adresstabelle anlegen
;------------------------------------
PROC Graphic_Init()
BYTE i

row=0
FOR i=0 TO 95
       DO
       adrl(i)=rowl
       adrh(i)=rowh
       row==+20
       OD
Zero(Vram1,Vrlen)
Zero(Vram2,Vrlen)
scr=0
Lms=Vram1
Wrk=Vram2
DJMP=@Dlst0
Dlist=@Dlst0
RETURN

;-----------------------------
;Graphik-Koerper zeichnen
;-----------------------------
PROC Draw(BYTE ARRAY Def,BYTE vergr,
                         INT xrel,yrel)
BYTE i
INT X,Y

IF Vergr=0 THEN RETURN FI
i=0
WHILE Def(i)<>$FF
       DO
       X=Def(i+1) X==+xrel-Mitte_X
       Y=Def(i+2) Y==+yrel-Mitte_Y
       X==*Vergr/10+79
       Y==*Vergr/10+48
       IF Def(i)=0 THEN
               Xnow=X Ynow=Y
       ELSE
               LineTO(X,Y)
       FI
       i==+3
       OD
RETURN

;------------------------------------
;Das Hauptprogramm:
;------------------------------------
PROC Vektorgraphik()
BYTE i
Graphic_Init()
DO
For i=0 TO VMax STEP 3
    DO
    Draw(ATARI_L,i,7,0)
    Draw(ATARI_L,Vmax-i,0-7,0)
    Switch_Screen()
    OD
For i=0 TO VMax STEP 3
    DO
    Draw(ATARI_L,Vmax-i,7,0)
    Draw(ATARI_L,i,0-7,0)
    Switch_Screen()
    OD
OD
RETURN
```

1050-Floppy mit Happy-Enhancement

Wie die Sache arbeitet und wie man die neuen Möglichkeiten nutzt, lesen Sie im ersten Teil unseres Kurses

Sicher weiß jeder, daß eine 1050-Diskettenstation, die mit einem Happy-Enhancement ausgerüstet ist, eine Vielzahl neuer Möglichkeiten bietet. Wie man diese nutzen kann, soll hier gezeigt werden.

Diese Station ist im Grunde ein kleiner Computer, der nur

8 Bit

dazu dient, die Daten sicher auf Diskette zu schreiben und zu lesen. Ihr Prozessor ist ein 6502, der mit 8-KByte-ROM und 8-KByte-RAM den Datenaustausch zwischen Computer und Floppy-Disk-Controler (FDC) regelt. Ferner ist noch ein RIOT 6532 enthalten, dessen Aufgaben später besprochen werden.

Aus mir unbekannten Gründen besteht das Floppy-ROM nicht aus einem 8-KByte-Block, sondern es ist in zwei 4-KByte-Blöcke unterteilt, die den Adreßbereich $f000 bis $ffff belegen. Das Umschalten zwischen beiden 4-KByte-Blöcken geschieht durch Load-Befehle:

LDA $fff8 Blendet den ersten 4-KByte-Block bei Adresse $f000 bis $ffff ein.

LDA $fff9 Blendet den zweiten 4-KByte-Block bei Adresse $f000 bis $ffff ein.

Für ein problemloses Umschalten zwischen beiden Blöcken werden gleich beim Einschalten der Floppy zwei Routinen ins RAM kopiert:

JSR $9600 Springt in den anderen
.WORD adr Block nach adr.
JSR $960A Ruft ein Unterprogramm
.WORD adr im anderen Block bei adr auf.

Wenn man vom ersten Block in den zweiten springen will, so muß die Zieladresse im zweiten um $8000 vermindert werden. Dies sagt den Routinen im RAM, aus welchem Block man kommt und in welchen man will. Das Unterprogramm zur Berechnung von Spur- und Sektornummer befindet sich z.B. in Block zwei bei Adresse $f7db. Will man es aus Block eins aufrufen, so schreibt man:

JSR $960A
.WORD $77db

Der Adreßbereich des RAM liegt bei $8000 bis $9fff, ferner auch von $0000 bis $01ff. Auf den Bereich zwischen $9800 und $9fff kann nicht ohne weiteres zugegriffen werden. Den RAM-Bereich zwischen $8000 und $97ff spricht man an, indem ein Read- oder Write-Kommando mit der RAM-Adresse als Sektornummer ausgeführt wird.

Der Floppy-Controler ist von Western Digital und hat die Bezeichnung WD 2793-PL 02. Er kann über folgende Register angesprochen werden:

$400 Kommando-Register (STA $400)
$400 Status-Register (LDA $400)
$401 Spur-Register
$402 Sektor-Register
$403 Daten-Register

Die einzelnen Kommandos des FDC:

$88 Lese Sektor
$a8 Schreibe Sektor
$c0 Lese Adresse
$e0 Lese Spur
$f0 Formatiere Spur
$d0 Erzwinge Interrupt

Der RIOT 6532 enthält einen kompletten PIA 6520, wie er auch im Atari enthalten ist (Joystickports). Dazu kommt noch ein Timer.

Die Adressen des RIOT:

$280 PORTA
$281 PACTL
$282 PORTB
$283 PBCTL

Die Funktion der einzelnen Bits:

PORTA:

7 1 FDC fordert Daten an.
0 FDC fordert keine Daten an
5 1 FDC im Single-Density-Modus
0 FDC im Double-Density-Modus
4 1 Schreib-Vorkompensation ein
0 Schreib-Vorkompensation aus
3 1 Motor aus
0 Motor ein

Bit 0 und 1 werden zur Festlegung der Laufwerknummer benützt.

PORTB:

7 1 Computer sendet gerade Kommando
0 Computer sendet kein Kommando
6 Dateneingang vom seriellen Bus

Bit 2 bis 5 werden für den Stepmotor benützt.

0 Datenausgang auf seriellen Bus

Da der FDC für bestimmte Kommandos (Formatiere Spur, Lese Spur und Lese Adresse) einen Index-Impuls vom Floppy-Disk-Interface benötigt und dieser Impuls nicht hardwaremäßig geliefert wird, verwendet man hierfür den Timer. Dieser gibt an den FDC einen Index-Impuls ab, sobald er auf Null abgelaufen ist. Die Register des Timers:

„Der Index-Impuls wird vom Timer geliefert"

$294 LADTIM Liefert Momentanwert des Timers
$296 FASTIM Ändert die Abzählgeschwindigkeit des Timers
$29f STATIM Setzt den Anfangswert des Timers

Unser erstes Ziel soll es sein, ein Listing des Floppy-ROM anzufertigen. Hierzu müssen wir der Floppy einen neuen Befehl beibringen. In ihrem RAM befindet sich bei Adresse $97a0 eine Tabelle, die alle Befehlssymbole, die der Floppy bekannt sind, enthält. Die Startadressen der zugehörigen ROM-Routinen sind in zwei weiteren Tabellen bei Adresse $97c0 (niederwertiges Byte der Startadresse) und $97e0 (höherwertiges Byte der Startadresse) enthalten. Bei einer Happy mit installiertem U.S.-Emulator sieht das folgendermaßen aus:

$97a0 .BYTE "PWpwRrS!"?$NOHQ"
Dann folgen 17 Nullen.
$97c0 .BYTE $17, $17, $12, $12, $76, $71, $93 bis ..., $82, $7a
$97E0 .BYTE $78, $78, $78, $78, $72, $72, $f7 bis ..., $76, $76

Man kann daraus ablesen, daß z.B. die Status-Routine in ROM-Block eins bei Adresse $f793 und die Q-Routine in ROM-Block zwei bei Adresse $f67a beginnt.

Wenn man der Floppy nun einen eigenen Befehl beibringen will, so muß man sein Symbol in die Tabelle ab Adresse $97a0 und seine Adresse in die Tabellen ab $97c0 und $97e0 schreiben. Platz für Ihre eigenen Befehle finden Sie im RAM von Adresse $8300 bis $95ff. Nun muß der Befehl ins RAM der Floppy übertragen werden, das noch vor Zugriff durch das Betriebssystem geschützt werden muß.

Eine Spur wird bei normaler Betriebsart auf einen Sitz in das RAM eingelesen, und jeder Sektor, der danach auf dieser Spur gelesen werden soll, wird direkt aus dem RAM zum Computer übertragen. Dies beschleunigt das Laden von Programmen um ca. 20%.

Ist die Floppy jedoch programmiert worden, so würde dies den Speicherinhalt wieder verändern und den neuen Befehl zerstören. Deshalb läßt sich das RAM durch den Befehl H (Status 0, DAUX $6060) vor einem Zugriff des Betriebssystems schützen.

Zu beachten ist außerdem, daß ein Befehl als Unterprogramm aufgerufen wird und somit über eine RTS-Instruktion verlassen werden muß.

Ein Programm, das alle diese Aufgaben übernimmt, finden Sie in Listing 1. Es geht davon aus, daß der neue Floppy-Befehl im Computer-RAM bei Adresse $9000-$93ff steht und in den gleichen Adreßbereich im Floppy-RAM gehört. Ferner gibt es dem neuen Befehl das Symbol X.

Listing 2 ist der neue Befehl X, der das Floppy-ROM ausliest. Um ihn zu verstehen, sehen wir uns zunächst einmal an, wie der Atari mit seinen Peripheriegeräten kommuniziert.

Sobald der Computer auf der Kommandoleitung einen Low-Impuls gibt (s. Belegung PORTB), wissen die Peripheriegeräte, daß ein sog. Command Frame folgt. Dies sind 5 Byte, die folgende Bedeutung haben:

1. Device I. D.
2. Kommando
3. DAUX
4. DAUX+1
5. Checksumme

Durch Device I. D. erkennt ein Peripheriegerät, ob es gemeint ist (für Diskettenstation 1 ist das z.B. $31). Kommando, DAUX und DAUX+1 sind genau die Bytes, die im Computer bei Adresse $302, $30a und $30b stehen. Sind Device I. D. $31 und die Checksumme in Ordnung, so sendet das Floppy-Betriebssystem eine $41 (Acknowledge) zum Computer, was bedeutet, daß der Befehl verstanden wurde. Nun holt es aus den Tabellen bei $97c0 und $97e0 die Anfangsadresse des entsprechenden Befehls und ruft diesen als Unterprogramm auf.

In DAUX und DAUX+1 ($82, $83) steht somit die Adresse, ab der 256 Byte des ROM ausgelesen werden sollen. Ist DAUX+1 positiv, so wird auf den zweiten 4-KByte-Block umgeschaltet und das oberste Bit von DAUX+1 gesetzt. Jetzt werden 256 Byte aus dem ROM ins RAM kopiert. Danach muß wieder auf den ersten 4-KByte-Block umgeschaltet werden. Nun wird dem Computer signalisiert, daß der Befehl abgeschlossen ist. Danach werden die 256 Byte zum Computer übertragen.

„Kommunikation zwischen Computer und Floppy"

Das Programm in Listing 3 liest mit Hilfe des neuen X-Befehls das Floppy-ROM aus. Wenn man alle drei Programme als Objekt-File auf Diskette vorliegen hat, so geht man hierzu folgendermaßen vor:

1. Ins DOS
2. Das Programm aus Listing 2 laden (Der neue Befehl)
3. Das Programm aus Listing 1 starten (Floppy wird programmiert)
4. Das Programm aus Listing 3 starten (Floppy-ROM wird in Computer geladen)
5. Den Speicher von $6000-$6fff als DROM1 abspeichern
6. Den Speicher von $7000-$7fff als DROM2 abspeichern

Diese zwei Blöcke müssen anschließend noch disassembliert und bearbeitet werden. Es folgt z.B. auf jeden JSR $9600 und JSR $960a eine 2-Byte-Adresse, die vom Disassembler sicher fälschlicherweise als Befehl interpretiert wird. Auch die Einführung von symbolischen Sprungadressen erleichtert das Verständnis des Floppy-Betriebssystems.

Zum Abschluß dieses Artikels möchte ich noch eine Reihe wichtiger ROM-Routinen und symbolischer Adressen nennen.

In Block 1:

$f000 Sende Puffer zum Computer
$f002 Sende Akku zum Computer
$f040 Neustart
$f1fb Motor aus
$f212 Starte Motor
$f239 Motor an
$f275 Kopf auf Spur 0
$f2ec Kopf auf Spur
$f362 Controler Reset
$f40b Kommando auswerten
$f485 Sende Acknowledge
$f48a Sende Nack
$f48f Sende Complete
$f494 Sende Error
$f499 Empfange 1 Byte vom Computer
$f5bb Lese einen Sektor
$f6a8 Schreibe einen Sektor
$f8d4 Formatiere eine Spur
$fba3 Motorblinken bei Fehler

In Block 2:

$f0f2 Density feststellen
$f37e Lese Spur ein
$f7db Spur und Sektor berechnen

$80 BUSID
$81 DKMD
$82 DAUX
$8d Spur
$99 Pufl
$9a Pufh
$8f Statusr
$B1 Chksum

$9600 Swijmp(Switch und jmp)
$960a Swijsr (Switch und jsr)
$96f7 Dsktyp
0: Enhanced Density
1: Double Density
$80: Single Density
$97a0 Ramkmdt
$97c0 Ramkmdl
$97e0 Ramkmdh

Im nächsten Heft soll der Umgang mit den einzelnen FDC-Befehlen beschrieben und nützliche neue Befehle für Ihre Happy vorgestellt werden.

Wer sich nicht die Arbeit machen will, ein eigenes ROM-Listing anzufertigen, kann es auch von mir (gegen Vorauszahlung von 10.– DM) erhalten. Die zwei Blöcke sind in Mac/65-Format abgespeichert.

Stefan Wachter
Haslacher Weg 45
7900 Ulm
Tel. 07 31 / 26 53 03

Listing 1

```
; PROGRAMMIERT DIE FLOPPY MIT
; DEM NEUEN X-BEFEHL, DER IM
; SPEICHER BEI BEFADR STEHEN MUSS
;
DSBI     =    $0300
DDRV     =    $0301
DKMD     =    $0302
DSTA     =    $0303
DPUF     =    $0304
DTIO     =    $0306
DLEN     =    $0308
DAUX     =    $030A
SIO      =    $E459
;
PUF      =    $0600
BEFTAB   =    $9780
BEFADR   =    $9000
;
         *=   $A000
;
; BEFEHLSTABELLE LADEN
;
         LDX # <PUF
         LDY # >PUF
         JSR SETPUF
         LDX # <BEFTAB
         LDY # >BEFTAB
         JSR SETAUX
         LDA #'R
         JSR DOSIO
;
; LEEREN BEFEHLSPLATZ SUCHEN
;
         LDY #$20
SUCH     LDA PUF,Y
         BEQ LEER
         INY
         JMP SUCH
;
; NEUEN BEFEHL EINFUEGEN
;
LEER     LDA #'X      ; NEUER BEFEHL
         STA PUF,Y    ; X
         LDA # <BEFADR
         STA PUF+$20,Y ; BEFEHLSADR.
         LDA # >BEFADR ; SETZEN
         STA PUF+$40,Y
;
; NEUE TABELLE ZUR FLOPPY
;
         LDA #'P
         JSR DOSIO

;
; NEUEN BEFEHL ZUR FLOPPY
; SCHICKEN
; ( COMPUTER-RAM  $9000-$93FF )
; ( IN FLOPPY-RAM $9000-$93FF )
;
         LDX # <BEFADR
         LDY # >BEFADR
         JSR SETPUF
         JSR SETAUX
PROG     LDA #'P
         JSR DOSIO
         CLC
         LDA DPUF
         ADC #$80
         STA DPUF
         BCC *+5
         INC DPUF+1
         LDA DPUF
         STA DAUX
         LDA DPUF+1
         STA DAUX+1   ; KOMMANDO KANN
         CMP #$94     ; BIS $93FF
         BCC PROG     ; GEHEN
;
; RAMBEREICH IN FLOPPY SCHUETZEN
;
         LDX #$60
         LDY #$60
         JSR SETAUX
         LDA #'H
         JSR DOSIO
;
; FERTIG, FLOPPY PROGRAMMIERT
;
         RTS
;
SETPUF   STX DPUF
         STY DPUF+1
         RTS
SETAUX   STX DAUX
         STY DAUX+1
         RTS
DOSIO    LDX #0
         CMP #'P
         BNE *+4
         LDX #$80
         CMP #'R
         BNE *+4
         LDX #$40
         STX DSTA
         STA DKMD
         LDA #'1

         STA DSBI
         LDA #1
         STA DDRV
         LDA #2
         STA DTIO
         LDA #128
         STA DLEN
         LDA #0
         STA DTIO+1
         STA DLEN+1
         JSR SIO
         BMI FEHLER
         RTS
;
FEHLER   LDA 710        ;BEI FEHLER
         PHA            ;HINTERGRUND-
         SBC #8         ;FARBE AENDERN
         STA 710
         LDA #0
         STA 20
WARTE    LDA 20         ;WARTEN
         CMP #50
         BCC WARTE
         PLA            ;ALTE HINTER-
         STA 710        ;GRUNDFARBE
         PLA
         PLA
         RTS
;
; STARTADRESSE
;
         *=  $02E0
         .WORD $A000

;
; SENDE 256 DATENBYTES MIT
; CHECKSUMME
;
         LDA # <RAMPUF
         STA PUFL
         LDA # >RAMPUF
         STA PUFH
```

```
        LDY #0
        JSR SENDPUF
;
; ZURUECK ZUM SYSTEM
;
        RTS
```

Listing 2

```
; DIESER BEFEHL ERMOEGLICHT DAS
; DAS AUSLESEN DES FLOPPYBETRIEB-
; SYSTEMS
;
DAUX    =     $82
PUFL    =     $99
PUFH    =     $9A
RAMPUF  =     $8000
;
SENDCPL =     $F48F
SENDPUF =     $F503
;
        *=    $9000     ; KOMMANDOADR.
;
        LDA DAUX+1  ; BEI POSITIVEM
        BMI NOSWITCH ; DAUX+1
        ORA #$80    ; 2. 4K ROM
        STA DAUX+1  ; EINBLENDEN
;
; BLENDE 2. 4K ROM EIN
;
        LDA $FFF9
;
NOSWITCH LDY #0
LOOP    CPY #$F8    ; AUFPASSEN,
        BCC OK      ; DASS NICHT
        CPY #$FA    ; AUS VERSEHEN
        BCS OK      ; UMGEBLENDET
        LDA DAUX+1  ; WIRD
        CMP #$FF    ; ( LDA $FFF8 )
        BNE OK      ; ( LDA $FFF9 )
        JMP WEITER
;
OK      LDA (DAUX),Y
        STA RAMPUF,Y
WEITER  INY
        BNE LOOP
;
; BLENDE 1. 4K ROM EIN
;
        LDA $FFF8
;
; SIGNALISIERE COMPUTER, DASS
; OPERATION BEENDET IST UND
; DATENBLOCK GLEICH FOLGT
;
        JSR SENDCPL
;
; DAS FLOPPY-ROM BEFINDET SICH
; JETZT IM COMPUTER SPEICHER
; 1. 4K BEI PUF1
; 2. 4K BEI PUF2
;
        RTS
;
DOSIO   LDA #'1
        STA DSBI
        LDA #1
        STA DDRV
        LDA #'X     ; DAS NEUE
        STA DKMD    ; KOMMANDO
        LDA #$40    ; STATUS READ
        STA DSTA
        LDA #2
        STA DTIO
        LDA #0
        STA DTIO+1  ; 256 BYTES
        STA DLEN    ; WERDEN VON
        LDA #1      ; DER FLOPPY
        STA DLEN+1  ; ERWARTET
        JSR SIO
        BMI FEHLER
        RTS
;
FEHLER  LDA 710     ; BEI FEHLER
        PHA         ; HINTERGRUND-
        SBC #8      ; FARBE AENDERN
        STA 710
        LDA #0
        STA 20
WARTE   LDA 20      ; WARTEN
        CMP #50
        BCC WARTE
        PLA         ; ALTE HINTER-
        STA 710     ; GRUNDFARBE
        PLA
        PLA
        RTS
;
; STARTADRESSE
;
        *=    $02E0
        .WORD $5000
```


Listing 3

```
; LAEDT DURCH DEN X-BEFEHL DAS
; FLOPPY BETRIEBSSYSTEM IN DEN
; COMPUTER
; ( 1. 4K NACH PUF1 )
; ( 2. 4K NACH PUF2 )
;
DSBI    =     $0300
DDRV    =     $0301
DKMD    =     $0302
DSTA    =     $0303
DPUF    =     $0304
DTIO    =     $0306
DLEN    =     $0308
DAUX    =     $030A
SIO     =     $E459
;
PUF1    =     $6000
PUF2    =     $7000
;
        *=    $5000
        LDA # <PUF1 ; 1. 4K ROM
        STA DPUF    ; NACH PUF1
        LDA # >PUF1
        STA DPUF+1
        LDA # <$F000 ; NEG. DAUX+1
        STA DAUX     ; SPRICHT
        LDA # >$F000 ; 1. 4K ROM
        STA DAUX+1   ; AN
;
LOOP1   JSR DOSIO    ; LADE 1. 4K ROM
        INC DPUF+1   ; IN COMPUTER
        INC DAUX+1
        LDA DAUX+1
        BNE LOOP1
;
        LDA # <PUF2  ; 2. 4K ROM
        STA DPUF     ; NACH PUF2
        LDA # >PUF2
        STA DPUF+1
        LDA # <$7000 ; POS. DAUX+1
        STA DAUX     ; SPRICHT
        LDA # >$7000 ; 2. 4K ROM
        STA DAUX+1   ; AN
;
LOOP2   JSR DOSIO    ; LADE 2. 4K ROM
        INC DAUX+1   ; IN COMPUTER
        INC DPUF+1
        LDA DAUX+1
        CMP #$80
        BNE LOOP2
```



TOP PRO-GRAMM DES MONATS

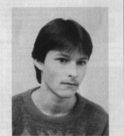
1000,-

Zum Topprogramm haben wir in diesem Heft das Programm "XL-TOS" von Marc Ebner aus Gerlingen bei Stuttgart ausgewählt. Es simuliert auf einem Atari 800 XL die GEM-Oberfläche wie beim ST. Damit können jetzt auch die 8-Bit-User so richtig mit GEM werkeln.

Marc Ebner ist 17 Jahre alt und besucht zur Zeit die 11. Klasse am Gerlinger Gymnasium. Seine ersten Erfahrungen sammelte er auf einem 600 XL, den er relativ schnell in Basic programmieren konnte. Danach folgte Assembler und bald darauf C. Im Sommer 1985 ging er dann für ein halbes Jahr auf eine Schule in Amerika und lernte Pascal. Da er sich dort aber mangels nötigem Kleingeld nur einen kleinen Atari leisten konnte, programmierte er sich einfach seine GEM-Oberfläche selbst.

Nach seiner Rückkehr aus den USA begann Marc dann im Februar 1986 auf dem ST zu programmieren. Nach einigen Utility-Programmen folgte die Arbeit an einem Karate-Spiel, das inzwischen fertig geworden ist. Inzwischen programmiert er an einem neuen Spiel, das sich an "The Last Starfighter" anlehnt.

Neben seiner großen Leidenschaft "Computer" geht Marc ganz gern ins Kino. Und was die Zukunft anbelangt, so will er nach dem Abitur Informatik studieren.



ATARI BUCHVERSAND



Lüke/Lüke
Der ATARI 520 ST

Dies ist das Buch für die erste Begegnung mit dem Atari ST. Die zweite Auflage dieses Standardwerks wurde überarbeitet und berücksichtigt nun die Neuerungen beim Betriebssystem und den Systemprogrammen. Daher wird diese allgemeine Beschreibung der Hard- und Software auch für die tägliche Arbeit ein hilfreicher Ratgeber sein.

Bestellnummer MT 23 **DM 49.–**



Geiß/Geiß
Logo auf dem Atari ST

Die Programmiersprache Logo erfreut sich steigender Beliebtheit. Nicht zuletzt deshalb, weil sie zum Lieferumfang des ST gehört. Daß Logo nicht nur für grafische Spielereien taugt, wird mit diesem Buch bewiesen: Textverarbeitung oder Mathematik sind ebenso möglich.

Bestellnummer HU 1 **DM 35.–**



Julian Reschke
Atari Basic Handbuch

208 Seiten
Das vorliegende Basic-Handbuch hilft Ihnen, Ihren Atari voll und ganz zu beherrschen. Das vollständige Basic-Vokabular wird beschrieben und anhand praktischer Beispiele erläutert.

Bestellnummer SY 13 **DM 32.–**



D. Senftleben
Start mit Atari-Logo

220 Seiten
Hier handelt es sich um eine benutzerfreundliche Einführung in die Computersprache Logo. Grafik, Text und Musik werden in zwölf Lektionen besprochen. Auch große Bildschirmfotos fehlen nicht. Die Atari-Logo-Vokabeln, die im Buch aufgeführt sind, erschließen dem Leser neue Einsatzbereiche.

Bestellnummer V 2 **DM 30.–**





L. M. Schreiber
Das Atari-Programmierhandbuch

390 Seiten
Hier werden keinerlei Kenntnisse vorausgesetzt. Sie lernen den Weg vom Problem zum Programm (einschließlich Flußdiagramm und dessen Gebrauch). Außerdem wird erklärt, wie Sie den 6502-Prozessor direkt programmieren. Wenn Sie dieses Buch durchgearbeitet haben, kennen Sie Ihren Atari in- und auswendig.

Bestellnummer MT 8 **DM 52.–**



Raabe/Schmidt
Spielen, lernen und arbeiten mit dem Atari

280 Seiten
Damit werden Ihnen theoretische und praktische Kenntnisse vermittelt. Von Anfang an lernen Sie Ihren Rechner Schritt für Schritt immer besser kennen und beherrschen. So werden Sie vom Spieler zum Profi.

Bestellnummer SY 14 **DM 32.–**



A. Hettinger/A. Heinz
Start mit Atari-Basic

184 Seiten
Nach dem Durcharbeiten dieses Buches werden Sie selbst in der Lage sein, Programme zu schreiben. Angefangen bei Grafik- und Soundmöglichkeiten über Tips und Tricks bis hin zu kompletten Spielprogrammen reicht das breite Spektrum. Neben dem eigentlichen Basic-Kurs bildet die komplett dokumentierte Liste aller Atari-Basic-Befehle die Krönung des Ganzen.

Bestellnummer V 3 **DM 30.–**



Dittrich
ATARI-ST – Peeks & Pokes

Wem Programmiersprachen und Anwenderprogramme auf dem ST nicht genügen, der ist mit diesem Buch gut bedient. Hier wird man hinter die Kulissen dieses außerordentlichen Computers geführt. Denn nur so können die wahren Dimensionen des Atari ST ausgelotet werden.

Bestellnummer DB 30 **DM 29.–**



Tom Rowley
Sprühende Ideen mit Atari Grafik

250 Seiten
Dies ist ein Lehrbuch, das mit den Grafikmöglichkeiten des Atari in die Gestaltgesetze von Objekten, in Farbgebung und in die Entwicklung von Bildschirmentwürfen einführt.

Bestellnummer TW 15 **DM 49.–**



A. + J. Peschetz
Was der Atari alles kann
Band 1

236 Seiten
Hier muß der Anwender schon die Grundbegriffe des Atari-Basic kennen und ein wenig Übung im Programmieren besitzen. Eine Vielzahl von gut durchstrukturierten Programmen aus den Bereichen Hobby, Wissenschaft, Beruf und Spiel werden vorgestellt.

Bestellnummer V 4 **DM 35.–**



Aumann/Maier/Stöpper
ATARI ST – Das Floppy-Arbeitsbuch

Die Diskettenstationen SF 354 und SF 314 sind Thema dieses Buches. Ausführlich wird auf die Programmierung im Diskettenbetrieb eingegangen. Beigelegt ist zusätzlich eine Diskette mit leistungsfähigen Programmen. So z.B. ein Kopierprogramm oder ein Disketten-Monitor.

Bestellnummer SY 29 **DM 69.–**



Naimann
ATARI-ST – Einführung in WordStar

WordStar gilt nach wie vor als der Klassiker der Textverarbeitung. Daher ist dieses Programm unter CP/M 2.2 auch für den Atari ST verfügbar. Dieses Buch ist eine Einführung in die Arbeit mit WordStar und wird dem Fortgeschrittenen als unentbehrliches Nachschlagewerk dienen.

Bestellnummer SY 30 **DM 48.–**



A. + J. Peschetz
Was der Atari alles kann
Band 2

240 Seiten
Entsprechend Band 1 enthält auch dieses Buch eine ausgewogene Mischung aus professionellen Anwendungsprogrammen und Spielen wie z.B. Dateiorganisation, Datensortiermethoden aber auch Trigonometrie in Verbindung mit deren ausgeklügelten Erläuterungen.

Bestellnummer V 5 **DM 35.–**



Stanley R. Trost
Atari-Programm-Sammlung

190 Seiten
Hier wird dem Anwender ein Satz ausgetesteter Programme für die Atari-Computer geboten. Eine breite Palette praktischer Beispiele hilft Ihnen, Ihren Computer optimal zu nutzen.

Bestellnummer SY 11 **DM 34.–**

Voss
Das Basic-Trainingsbuch zu Atari 600 XL/800 XL

383 Seiten
Das Basic-Trainingsbuch zu Atari 600 XL/800 XL ist eine ausführliche, didaktisch gut geschriebene Einführung in das Atari-Basic. Von den Befehlen über die Problemanalyse bis zum fertigen Algorithmus lernt man schnell das Programmieren.

Bestellnummer DB 17 **DM 39.–**



Löhr
Assembler-Praxis auf Atari ST

Die Programmierung des 68000-Mikroprozessors auf dem Atari ST verlangt nicht nur genaue Kenntnisse des System, auch der Umgang mit Assemblern oder Editoren will gelernt sein. Diesen Gesichtspunkt der praktischen Anwendung stellt der Autor des Buches in den Mittelpunkt, wobei auch die Grundlagen keineswegs zu kurz kommen.

Bestellnummer TW 16 **DM 59.–**





Reschke/Wiethoff
Das Atari Profibuch

320 Seiten
In diesem Werk finden Sie gebündelt alle wichtigen Informationen, um Ihren Atari genau kennenzulernen und seine Fähigkeiten voll auszureizen. Ein Informationspaket, das keine Fragen offen läßt.

Bestellnummer SY 12 **DM 42.–**



Don Inman/Kurt Inman
Der Atari Assembler

276 Seiten
Mit diesem Buch können Sie das Programmieren in Assembler lernen und sich gleichzeitig mit der Anwendung des Atari-Assembler-Moduls auf Ihrem Atari 400- oder 800-Modell vertraut machen.

Bestellnummer ID 18 **DM 36.–**



Grohmann/Seidler/Slibar
Das Maschinensprachebuch zum ATARI ST

Der Mikroprozessor 68000 verleiht dem Atari ST seine sprichwörtliche Leistungsfähigkeit. Dieses Buch ist ein Lehrbuch zur Programmierung in Maschinensprache. Auch ohne Vorkenntnisse bekommt man damit bald direkten Zugang zu den unglaublichen Fähigkeiten dieses Prozessors.

Bestellnummer DB 28 **DM 39.–**



C. Lorenz
Das große Spielebuch für Atari, Band 1

151 Seiten
Aufregende Computerspiele in Atari-Basic. Neben Spielen finden Sie hier eine Reihe hochinteressanter Anregungen für eigene Programme. 3-D-Grafik, Bewegung und Scrollen, Grafik und Ton in Forth, Tonprogrammierung usw.

Bestellnummer HO 25 **DM 29.80**



Poole/McNiff/Cook
Mein Atari-Computer

500 Seiten
Ein Handbuch, das für jeden Atari-Besitzer wertvolle Informationen enthält und zur Lösung aller Atari-Probleme beiträgt. Es ist reich bebildert und enthält eine Vielzahl der für den ernsthaften Interessierten so wichtigen Tabellen.

Bestellnummer TW 20 **DM 59.–**



C. Lorenz
Das große Spielebuch für Atari, Band 2

200 Seiten
Dieses Buch enthält Programme für den Atari 600 XL/800 XL und ist eine Weiterführung von Band 1. Es bringt eine Reihe neuer Spiele, Programme zur Sounderzeugung und ein Kapitel über Grafik-Spielereien.

Bestellnummer HO 26 **DM 29.80**



Steiner/Steiner
GEM für den Atari 520 ST

344 Seiten
Dieses Werk ist eine Einweisung in alle Bereiche, die GEM für den Benutzer interessant machen. Der unerfahrene Anwender findet eine Menge Tips für die Bedienung, um effektiv mit dem Atari ST arbeiten zu können.

Bestellnummer MT 21 **DM 52.–**





Walkowiak
Adventures, und wie man sie auf dem Atari 600 XL/800 XL programmiert

284 Seiten
Hier wir gezeigt, wie Adventures funktionieren, wie man sie erfolgreich spielt, und wie man eigene Adventures auf Atari-Computern der Serie XL programmiert. Hinzu kommt ein kompletter Adventure-Generator, der das Selberprogrammieren zum Kinderspiel macht.

Bestellnummer DB 27 **DM 39.–**

Jürgensmeier
WordStar für den Atari ST

435 Seiten
Dieses Buch ist so aufgebaut, daß der Leser mit WordStar schrittweise vertraut wird. Anhand von Beispielen werden alle Funktionen erläutert. Auch auf die Bedienung von MailMerge wird ausführlich eingegangen.

Bestellnummer MT 22 **DM 49.–**



Schwaiger
Atari Star-Texter

110 Seiten + Disk
Hierbei handelt es sich um eine umfangreiche, komfortable Textverarbeitung für Ihren Atari (mind. 48 KByte). Das Buch gibt eine Einführung, die Diskette bietet ein exzellentes Programm.

Bestellnummer SY 28 **DM 64.–**



Severin
Das große DFÜ-Buch zum ATARI ST

Die Verbindung des Atari ST mit der weiten Welt der Mailboxen und Datenbanken wird in diesem Buch mit allen Aspekten abgehandelt. Von den Grundlagen über ein komplettes Mailboxprogramm bis zur Telefonliste bietet es eine fundierte Einführung in die DFÜ.

Bestellnummer DB 29 **DM 39.–**

Rugg/Feldman/Barry
30 Basic-Programme für den Atari

274 Seiten
Das Buch enthält sorgfältig getestete Spiel- und Grafikprogramme aus Mathematik, Unterricht und vielen anderen Anwendungsbereichen des täglichen Lebens für Ihren Atari-Computer.

Bestellnummer ID 29 **DM 34.–**



Alfred Görgens
Utilities in Basic für Atari-Computer

120 Seiten
In diesem Buch finden Sie praktische Utilities zu den Themen Programmierhilfe, Sound und Textverarbeitung. So z.B. automatische Zeilennumerierung, Umnumerierung von Basic-Zeilen, automatischer Programmstart, Musikeditor oder auch die Wiedergabe von Atari-Zeichen und Musiknoten auf dem Drucker.

Bestellnummer V 24 **DM 25.–**



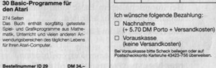
James/Gee/Ewbank
Das Atari Spielebuch für 600 XL/800 XL

184 Seiten
21 Spiele vermitteln Spannung, Action und bewegte Grafik. Jedes Programm ist vollständig aufgelistet und ausführlich erläutert. So wird auch der Newcomer mit der Syntax und dem Aufbau der Programmiersprache Basic vertraut gemacht. Er lernt die Routine verstehen, analysieren und kann sie somit auch in eigene Programme einbinden.

Bestellnummer V 30 **DM 30.–**

Buch-Bestellschein

Bitte liefern Sie mir folgende Bücher:



Anzahl	Bestell-Nr.	Einzel-Preis

Ich wünsche folgende Bezahlung:

☐ Nachnahme
(+ 5.70 DM Porto + Versandkosten)

☐ Vorauskasse
(keine Versandkosten)

Bei Vorauskasse bitte Scheck beilegen oder auf Postscheckkonto Karlsruhe 43423-756 überweisen.

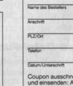
Name des Bestellers

Anschrift

PLZ/Ort

Telefon

Datum/Unterschrift



Coupon ausschneiden, auf Postkarte kleben und einsenden: ATARI magazin, Postfach 1640, 7518 Bretten.

XL-TOS

Das XL-TOS ersetzt die Disk Utility Package (DUP.SYS) völlig. Es wird stattdessen auf die Systemdiskette kopiert. Beim Booten erscheint zunächst das leere Desktop, das der Oberfläche des Atari ST nachempfunden ist. Der Pfeil, mit dem alle Befehle angesprochen werden, wird mit dem Joystick gesteuert.

In der Menüleiste gibt es vier Pull-Down-Windows mit den Bezeichnungen Desk, File, View und Options. Auf dem Desktop sind drei Icons (Sybole für die entsprechenden Befehle) vorhanden: die zwei Disketten-Icons, um das Directory von Laufwerk A bzw. B aufzurufen, und ein Mülleimer-Icon.

Die Pull-Down-Windows spricht man durch einmaliges Anklicken von Desk, File, View bzw. Options an. Danach öffnet sich ein Drop-Down-Window, in dem man die weiteren Funktionen durch Anklicken aufrufen kann. Es kann nur ein Pull-Down-Window geöffnet werden; ein altes ist zuvor zu schließen. Hierzu klickt man das Close-Feld in der linken oberen Ecke des Windows an. Die Pfeile dienen dazu, den Inhalt, sofern er nicht hineinpaßt, zu scrollen. Dies ist z.B. oft beim Disketten-Directory der Fall. Die Kopfzeile ist ohne Funktion und gibt lediglich Informationen über das Window.

Bei Dialog-Windows verschwindet der Pfeil, und der Cursor erscheint. Es wird eine Eingabe erwartet. Zur Sicherheit wird die selektierte Funktion nochmals angegeben. Man kann diese Windows durch Drücken der Return-Taste verlassen.

Desk (allgemeine Funktionen)

Basic: Falls ein Basic beim Booten des TOS vorhanden war, kann durch diesen Menü-Eintrag dahin zurückgesprungen werden. Bei ausgeschaltetem Basic kann man es natürlich nicht mehr aufrufen.

Boot: Die Diskette, die im Laufwerk A ist, wird gebootet.

Info: Ein Window mit einer kurzen Information über das Programm wird geöffnet. Nach einmaligem Drücken des Joystick-Knopfes wird das Window wieder geschlossen.

File (Disketten-Datei-Funktionen)

Erase: Ein Dialog-Window öffnet sich und fragt nach dem zu löschenden Filenamen. Die Laufwerknummer "D1:" kann weggelassen werden.

Protect: Files können mit dieser Funktion geschützt werden.

Unprotect: Files werden wieder zur Bearbeitung freigegeben.

Rename: Files können umbenannt werden.

Format SD: Die Diskette kann mit 707 freien Sektoren formatiert werden. Die Diskettenstation muß im Format "Dx:" angesprochen werden, wobei x für die Laufwerknummer steht.

Format DD: Unter DOS 2.5 kann man die Disketten mit 1040 Sektoren formatieren.

View (Darstellung ändern)

Directory on Screen: Das Directory wird von nun an nicht mehr im Window gezeigt, sondern auf dem ganzen Bildschirm wie beim normalen DUP. SYS.

Directory in Window: Dies wird beim Aufruf selektiert. Das Directory erscheint im Window.

Options (Optionen einstellen)

Request on: Wird ebenfalls beim Aufruf selektiert. Es wird nachgefragt, ob der Befehl ausgeführt werden soll oder nicht. Es wird jedesmal ein Alarm-Window mit zwei Knöpfen geöffnet. Der eine dient zum Fortfahren ("Cont"), der andere zum Abbrechen ("Cancel"). Nun muß man einen von beiden Knöpfen betätigen.

Request off: Ist dieser Eintrag selektiert, so wird nicht mehr nachgefragt.

Icons

Diskette A: Es wird ein Directory von Diskette A angezeigt.

Diskette B: Es wird ein Directory von Diskette B angezeigt.

Mülleimer: Die Schrift Trash wird selektiert. Klickt man jetzt einen Programmnamen im Window an, so wird er gelöscht. Durch einen Klick auf das Desktop wird er deselektiert.

Programmnamen: Das angeklickte Programm wird gestartet.

Bei einem Fehler erscheint ein Alarm-Window mit zwei Knöpfen und der Fehlernummer. Mit Strt wird das TOS neu gestartet, mit Quit wird die Diskette in Laufwerk A gebootet. Das TOS kann auf jede Diskette kopiert werden, auf der DOS.SYS vorhanden ist.

Marc Ebner


```
1 REM XL-TOS by
2 REM
3 REM Marc Ebner
4 REM Karlstrasse 50
5 REM 7016 Gerlingen
6 REM Telefon: 07156 / 28665
7 REM
9 TRAP 200
10 OPEN #1,8,0,"D:DUP.SYS"
20 ZEILE=1000:READ NUM
30 FOR A=1 TO 17:Z=Z+1
40 READ B:? #1;CHR$(B);
50 SUM=SUM+B:IF Z=NUM-1 THEN 100
60 NEXT A
70 READ C:IF C<>SUM THEN 200
80 ZEILE=ZEILE+10:SUM=0
90 GOTO 30
100 READ C:IF C<>SUM THEN 200
110 ? "DONE !":END
200 ? "FEHLER IN ZEILE ";ZEILE
210 END
900 DATA 6243
1000 DATA 255,255,25,6,27,6,68,49,58,110,6,116,6,68,49,58,42,1204
1010 DATA 46,42,155,0,32,251,32,32,114,39,32,235,40,32,148,40,32,1302
1020 DATA 53,41,32,96,41,32,118,41,32,163,41,32,14,42,76,9,32,895
1030 DATA 32,82,39,240,1,96,169,1,32,183,36,32,177,33,32,151,32,1368
1040 DATA 169,0,141,119,32,169,0,141,239,32,169,32,141,240,32,162,3,1821
1050 DATA 160,4,32,60,33,32,169,32,173,117,32,141,111,32,173,118,32,1451
1060 DATA 141,112,32,32,169,32,173,117,32,141,113,32,173,118,32,141,114,1704
1070 DATA 32,32,241,32,32,22,33,32,120,32,32,151,32,76,66,32,0,997
1080 DATA 0,0,0,0,0,0,0,0,173,226,2,141,133,32,173,227,2,1109
1090 DATA 141,134,32,32,255,255,173,224,2,141,148,32,173,225,2,141,149,2259
1100 DATA 32,32,255,255,96,169,168,141,226,2,141,224,2,169,32,141,227,2312
1110 DATA 2,141,225,2,96,96,162,16,169,7,157,66,3,169,117,157,68,1653
1120 DATA 3,169,32,157,69,3,169,2,157,72,3,169,0,157,73,3,32,1270
1130 DATA 86,228,48,37,173,117,32,201,255,208,7,173,118,32,201,255,240,2411
1140 DATA 210,173,119,32,208,17,173,117,32,141,239,32,173,118,32,141,240,2197
1150 DATA 32,169,255,141,119,32,96,76,255,255,173,113,32,56,237,111,32,2184
1160 DATA 141,115,32,173,252,32,247,33,114,32,237,112,32,141,116,32,173,2014
1170 DATA 115,32,24,105,1,141,115,32,173,116,32,105,0,141,116,32,96,1376
1180 DATA 162,16,169,7,157,66,3,173,111,32,157,68,3,173,112,32,157,1598
1190 DATA 69,3,173,115,32,157,72,3,173,116,32,157,73,3,32,86,228,1524
1200 DATA 32,19,35,96,138,162,16,157,66,3,173,29,6,201,58,240,20,1451
1210 DATA 173,30,6,201,58,240,13,169,25,157,68,3,169,6,157,69,3,1547
1220 DATA 76,103,33,169,28,157,68,3,169,6,157,69,3,152,157,74,3,1427
1230 DATA 32,86,228,32,19,35,96,32,78,46,32,160,46,32,216,46,32,1248
1240 DATA 105,47,32,174,47,32,142,33,165,166,208,3,76,27,32,76,253,1618
1250 DATA 33,162,0,189,28,6,201,0,240,10,24,105,32,157,28,6,232,1453
1260 DATA 76,144,33,169,155,157,28,6,96,160,95,162,228,169,6,32,92,1808
1270 DATA 228,96,32,167,33,169,0,141,29,208,169,0,133,84,133,85,162,1869
1280 DATA 109,160,34,32,167,43,169,224,141,244,2,169,0,141,240,2,169,2046
1290 DATA 0,133,84,133,85,162,241,160,33,32,167,43,169,0,133,84,169,1828
1300 DATA 12,133,85,162,25,160,6,32,167,43,169,2,133,82,96,76,111,1494
1310 DATA 97,100,105,110,103,248,33,243,34,32,61,61,62,155,32,82,39,1597
1320 DATA 240,1,96,169,2,32,183,36,162,33,160,0,32,60,33,162,16,1417
1330 DATA 32,99,39,32,82,42,96,169,5,32,183,36,169,0,133,84,133,1366
1340 DATA 85,162,109,160,34,32,167,43,162,3,160,4,32,60,33,32,191,1469
1350 DATA 34,169,1,141,254,2,32,227,34,48,9,32,146,43,32,163,34,1401
1360 DATA 76,57,34,192,136,240,3,76,19,35,162,134,160,34,32,167,43,1600
1370 DATA 173,132,2,208,251,32,90,41,141,254,2,162,16,32,99,39,162,1836
1380 DATA 32,32,99,39,76,0,32,156,156,156,156,156,156,156,156,156,156,1870
1390 DATA 156,156,156,156,156,156,156,156,156,156,156,156,156,156,155,80,114,253
3
1400 DATA 101,115,115,32,115,116,105,99,107,32,98,117,116,116,111,110,32,1637
1410 DATA 116,111,32,103,111,32,111,110,46,155,166,167,240,23,72,162,32,1789
```

```
1420 DATA 169,11,157,66,3,169,0,157,72,3,157,73,3,104,32,86,228,1490
1430 DATA 32,19,35,96,165,167,240,28,162,32,169,3,157,66,3,169,224,1767
1440 DATA 157,68,3,169,34,157,69,3,169,8,157,74,3,32,86,228,32,1449
1450 DATA 19,35,96,80,58,155,162,16,169,7,157,66,3,169,0,157,72,1421
1460 DATA 3,157,73,3,32,86,244,34,239,35,228,96,169,255,133,166,169,2122
1470 DATA 0,133,85,169,21,133,84,162,10,160,35,32,167,43,96,32,32,1394
1480 DATA 32,212,242,225,243,232,155,48,1,96,132,165,32,51,35,32,71,2004
1490 DATA 35,32,53,41,32,96,41,32,118,41,32,163,41,32,47,36,32,904
1500 DATA 90,36,76,30,35,169,0,133,128,162,8,160,4,169,23,133,142,1498
1510 DATA 169,15,133,140,32,18,43,96,169,10,133,85,133,82,169,6,133,1566
1520 DATA 84,162,143,160,35,169,154,133,168,169,0,133,169,32,154,36,164,2065
1530 DATA 165,132,212,169,0,133,213,32,170,217,32,230,216,160,0,177,243,2501
1540 DATA 48,8,9,128,153,41,36,200,208,244,153,41,36,169,18,133,85,1710
1550 DATA 169,8,133,84,162,41,160,36,32,167,43,96,65,110,32,101,114,1553
1560 DATA 114,111,114,32,110,117,109,98,101,114,155,155,155,155,104,97,115,1956
1570 DATA 32,111,99,99,117,114,101,100,46,32,80,108,101,97,115,101,155,1608
1580 DATA 99,104,101,99,107,32,121,111,117,114,32,115,121,115,116,101,109,1714
1590 DATA 46,155,81,117,105,116,32,97,110,100,32,114,101,98,111,111,116,1642
1600 DATA 32,116,104,101,155,115,121,115,116,101,109,44,32,111,114,32,114,1632
1610 DATA 101,115,116,97,114,116,63,240,35,235,36,155,155,155,32,32,18,1815
1620 DATA 16,16,16,16,19,32,32,32,18,16,16,16,16,19,155,32,32,499
1630 DATA 15,81,117,105,116,15,32,32,32,15,83,116,114,116,15,155,32,1191
1640 DATA 32,20,16,16,16,16,21,32,32,32,20,16,16,16,16,21,155,493
1650 DATA 32,32,32,32,32,155,162,78,160,36,169,5,32,125,36,208,3,1329
1660 DATA 160,81,96,162,84,160,36,169,5,32,125,36,208,3,160,83,96,1696
1670 DATA 160,78,96,79,49,117,105,116,79,79,51,116,114,116,79,192,78,1704
1680 DATA 208,1,96,192,81,208,3,76,119,228,162,64,32,99,39,162,32,1802
1690 DATA 32,99,39,162,16,32,99,39,169,0,141,29,208,76,0,32,142,1315
1700 DATA 144,36,140,145,36,133,131,162,255,232,228,131,240,11,189,28,6,2247
1710 DATA 221,255,255,240,243,160,255,96,160,0,96,138,162,0,157,68,3,2509
1720 DATA 152,157,69,3,169,11,157,66,3,165,168,157,72,3,165,169,157,1843
1730 DATA 73,3,32,86,228,96,133,174,173,160,49,201,23,208,41,32,156,1868
1740 DATA 38,32,17,39,32,234,36,32,254,36,32,53,41,32,96,41,32,1077
1750 DATA 118,41,32,163,41,32,89,38,32,136,38,48,3,76,204,36,32,1159
1760 DATA 180,38,32,38,39,96,169,0,236,36,231,37,133,128,162,2,160,1717
1770 DATA 8,169,21,133,142,169,8,133,140,32,18,43,96,169,4,133,85,1503
1780 DATA 133,82,169,10,133,84,32,82,37,162,27,160,37,169,55,133,168,1673
1790 DATA 169,0,133,169,32,154,36,96,155,18,16,16,16,16,16,16,19,1077
1800 DATA 32,18,16,16,16,16,16,16,19,155,15,67,97,110,99,101,108,917
1810 DATA 15,32,15,22,67,111,110,116,22,15,155,20,16,16,16,16,16,780
1820 DATA 16,21,32,20,16,16,16,16,16,16,21,155,165,174,201,100,144,1145
1830 DATA 3,56,233,100,10,10,10,10,170,134,162,166,162,189,153,37,201,1806
1840 DATA 155,240,8,32,146,43,230,162,76,98,37,32,146,43,165,174,201,1988
1850 DATA 9,176,28,173,29,6,201,58,240,14,173,30,6,201,58,240,7,1649
1860 DATA 162,25,160,6,76,149,37,162,28,160,6,32,167,43,96,70,111,1490
1870 DATA 114,109,97,116,32,83,68,32,100,105,115,107,58,155,82,117,110,1600
1880 DATA 32,102,105,108,101,58,32,32,32,32,32,32,155,69,114,97,115,1248
1890 DATA 101,32,102,105,108,101,58,32,32,32,32,155,80,114,111,116,101,1412
1900 DATA 99,116,32,102,105,108,101,58,32,32,155,85,110,112,114,111,116,1588
1910 DATA 101,99,116,32,102,105,108,101,58,232,37,227,38,155,67,111,112,1801
1920 DATA 121,32,102,105,108,101,58,32,32,32,32,32,155,82,101,110,97,1332
1930 DATA 109,101,32,102,105,108,101,58,32,32,32,155,68,105,114,101,99,1454
1940 DATA 116,111,114,121,58,32,32,32,32,32,155,70,111,114,109,97,116,1452
1950 DATA 32,68,68,32,100,105,115,107,58,155,32,32,32,32,32,32,32,1064
1960 DATA 32,32,32,32,32,32,32,32,155,82,117,110,32,99,97,114,116,1178
1970 DATA 114,105,100,103,101,32,32,155,66,111,111,116,32,110,101,119,32,1540
1980 DATA 100,105,115,107,32,32,155,162,120,160,38,169,7,32,125,36,208,1703
1990 DATA 3,160,81,96,162,128,160,38,169,7,32,125,36,208,3,160,67,1635
2000 DATA 96,160,78,96,79,35,97,110,99,101,108,79,79,86,35,111,110,1559
2010 DATA 116,86,79,192,78,208,3,160,0,96,192,67,208,3,160,255,96,1999
2020 DATA 32,180,38,76,9,32,162,0,160,144,32,211,38,96,162,0,160,1532
2030 DATA 148,32,211,38,96,162,0,160,152,32,211,38,96,162,0,160,144,1842
2040 DATA 32,226,38,96,162,0,160,148,32,226,38,165,178,240,3,32,180,1956
2050 DATA 45,96,162,0,160,152,32,226,38,96,134,172,132,173,165,88,133,2004
2060 DATA 170,165,89,133,171,76,238,38,134,170,228,38,98,39,132,171,165,2255
```



```
2070 DATA 88,133,172,165,89,133,173,169,0,133,85,133,84,169,32,32,146,1936
2080 DATA 43,162,0,160,0,177,170,145,172,200,208,249,230,171,230,173,232,2722
2090 DATA 224,4,208,238,32,55,39,96,162,0,189,28,6,157,68,6,201,1713
2100 DATA 155,240,8,201,0,240,4,232,76,19,39,96,162,0,189,68,6,1735
2110 DATA 157,28,6,201,155,240,4,232,76,40,39,96,165,166,240,7,162,2014
2120 DATA 10,160,35,76,70,39,162,110,160,40,169,0,133,85,169,21,133,1572
2130 DATA 84,32,167,43,96,173,29,6,201,48,144,4,201,91,144,3,160,1626
2140 DATA 255,96,160,0,96,99,39,94,40,169,12,157,66,3,32,86,228,1632
2150 DATA 96,169,96,141,244,2,96,32,122,40,162,0,32,99,39,162,0,1532
2160 DATA 169,3,157,66,3,169,119,157,68,3,169,40,157,69,3,169,12,1533
2170 DATA 157,74,3,32,86,228,32,108,39,169,1,141,240,2,162,0,169,1643
2180 DATA 11,157,66,3,169,186,157,68,3,169,39,157,69,3,169,189,157,1772
2190 DATA 72,3,169,0,157,73,3,32,86,228,96,125,32,32,32,68,101,1309
2200 DATA 115,107,32,32,70,105,108,101,32,32,86,105,101,119,32,32,79,1288
2210 DATA 112,116,105,111,110,115,155,17,17,17,17,17,17,17,17,17,17,994
2220 DATA 17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,17,289
2230 DATA 17,17,17,17,17,17,17,17,17,17,17,17,17,155,155,155,32,718
2240 DATA 32,32,32,32,32,32,32,65,155,32,32,32,32,32,1,2,155,762
2250 DATA 32,32,32,32,32,3,4,155,32,70,108,111,112,112,121,32,68,1088
2260 DATA 105,115,107,155,155,155,32,32,32,32,32,32,32,32,66,155,32,1301
2270 DATA 32,32,32,32,1,2,155,32,32,32,32,32,3,4,155,32,70,710
2280 DATA 108,111,112,112,121,32,68,105,115,107,155,155,155,155,32,32,32,1707
2290 DATA 32,32,5,6,155,32,95,40,90,41,32,32,32,32,7,8,155,826
2300 DATA 32,32,32,32,32,9,10,155,32,32,32,84,114,97,115,104,155,1099
2310 DATA 69,58,155,169,100,133,156,133,157,169,0,133,82,133,175,133,176,2131
2320 DATA 133,178,133,179,169,0,133,166,32,55,39,96,162,40,160,158,169,2002
2330 DATA 6,32,92,228,96,173,120,2,133,164,41,1,208,2,198,157,165,1818
2340 DATA 164,41,2,208,2,230,157,165,164,41,4,208,2,198,156,165,164,2071
2350 DATA 41,8,208,2,230,156,165,156,141,0,208,164,157,162,0,189,224,2211
2360 DATA 40,201,255,240,7,145,148,232,200,76,202,40,169,1,141,30,208,2335
2370 DATA 76,95,228,0,0,56,48,56,44,6,2,0,0,255,169,0,133,1168
2380 DATA 146,133,148,133,150,133,152,133,154,169,115,133,147,169,116,133,149,24
13
2390 DATA 169,117,133,151,169,118,133,153,169,119,133,155,169,112,141,7,212,2360
2400 DATA 169,62,141,47,2,169,3,141,29,208,169,0,168,145,148,145,146,1892
2410 DATA 145,150,145,152,145,154,200,208,243,169,1,141,111,2,169,0,141,2276
2420 DATA 192,2,96,173,132,2,208,251,32,90,41,76,89,41,169,0,133,1727
2430 DATA 20,173,120,2,201,15,208,234,165,20,201,10,240,228,173,132,2,2144
2440 DATA 208,238,32,90,41,96,173,91,41,86,42,132,2,240,251,96,165,2024
2450 DATA 156,56,233,44,74,74,24,133,159,165,157,56,233,28,74,74,74,1814
2460 DATA 24,133,158,96,165,88,133,138,165,89,133,139,162,0,228,158,240,2249
2470 DATA 17,165,138,24,105,40,133,138,165,139,105,0,133,139,232,76,128,1877
2480 DATA 41,165,138,24,101,159,133,138,165,139,105,0,133,139,96,160,0,1836
2490 DATA 177,138,141,108,6,177,138,201,0,240,16,165,138,56,233,1,133,2068
2500 DATA 138,165,139,233,0,133,139,76,170,41,165,138,24,105,1,133,138,1938
2510 DATA 165,139,105,0,133,139,162,0,177,138,157,28,6,201,0,240,17,1807
2520 DATA 165,138,24,105,1,133,138,165,139,105,0,133,139,232,76,207,41,1941
2530 DATA 173,28,6,201,67,240,7,169,39,133,145,76,251,41,169,79,133,1957
2540 DATA 145,165,138,56,229,145,133,138,165,139,233,0,133,139,177,138,141,2414
2550 DATA 109,6,96,173,108,6,201,75,240,46,201,76,240,45,201,78,240,2141
2560 DATA 44,173,28,6,201,65,240,28,201,67,240,24,201,69,240,35,201,2063
2570 DATA 71,240,31,201,73,240,27,201,33,144,4,201,59,144,16,32,140,1857
2580 DATA 40,96,76,82,42,76,192,43,76,16,46,76,45,46,76,114,33,1175
2590 DATA 76,246,34,173,109,6,201,34,87,42,82,43,208,11,169,50,141,1712
2600 DATA 111,6,141,26,6,76,108,42,169,49,141,111,6,141,26,6,173,1338
2610 DATA 43,49,201,23,240,19,32,229,42,32,246,43,162,64,32,99,39,1595
2620 DATA 169,0,133,137,32,180,45,96,169,10,133,82,162,209,160,42,32,1791
2630 DATA 167,43,32,70,44,32,99,44,48,10,162,0,160,6,32,167,43,1159
2640 DATA 76,148,42,162,0,160,6,32,167,43,162,211,160,42,32,167,43,1653
2650 DATA 173,132,2,208,251,32,90,41,162,64,32,99,39,162,32,32,99,1650
2660 DATA 39,162,16,32,99,39,169,0,141,29,208,76,0,32,96,125,155,1418
2670 DATA 80,114,101,115,115,32,70,105,114,101,32,66,117,116,116,111,110,1615
2680 DATA 155,169,0,133,177,165,178,208,13,169,0,133,179,165,175,240,2,2261
2690 DATA 230,179,32,172,38,169,255,133,178,169,3,133,128,162,15,160,5,2161
2700 DATA 169,22,133,142,169,13,133,140,32,18,43,96,134,82,134,85,134,1679
```

```
2710 DATA 143,132,144,136,132,84,169,155,32,146,43,169,11,32,146,43,169,1886
2720 DATA 16,32,124,43,169,12,32,146,43,169,155,32,146,43,169,15,32,1378
2730 DATA 146,43,169,32,32,124,43,169,15,32,146,43,169,155,32,146,43,1539
2740 DATA 198,140,165,140,208,230,169,13,32,83,43,78,44,146,43,169,16,1917
2750 DATA 32,124,43,169,14,32,146,43,169,155,32,146,43,165,128,240,19,1700
2760 DATA 165,144,24,101,128,133,84,165,143,24,105,1,133,85,169,17,32,1653
2770 DATA 124,43,96,133,163,165,142,133,141,198,141,198,141,165,163,32,146,2324
2780 DATA 43,198,141,165,141,208,245,96,72,162,0,169,11,157,66,3,169,2046
2790 DATA 0,157,72,3,157,73,3,104,32,86,228,96,169,9,141,66,3,1399
2800 DATA 142,68,3,140,69,3,162,0,169,255,157,72,3,157,73,3,32,1508
2810 DATA 86,228,96,165,158,201,5,208,24,169,255,133,177,165,179,208,6,2463
2820 DATA 169,0,133,175,133,176,169,0,133,178,133,179,32,203,38,96,169,2116
2830 DATA 0,133,177,165,176,208,6,169,0,133,178,133,179,169,0,133,175,2134
2840 DATA 133,176,32,188,38,96,169,0,133,129,169,128,133,130,160,0,169,1983
2850 DATA 32,145,129,200,208,249,230,130,165,130,201,132,240,3,76,0,44,2314
2860 DATA 169,0,133,129,169,128,133,130,169,0,133,132,133,133,32,70,44,1837
2870 DATA 32,99,44,48,19,32,20,45,165,129,24,105,2,133,129,165,130,1321
2880 DATA 105,0,133,130,76,35,44,32,51,45,165,132,56,233,8,133,134,1512
2890 DATA 96,162,64,169,3,157,66,3,169,110,79,44,74,45,157,68,3,1469
2900 DATA 169,6,157,69,3,169,6,157,74,3,32,86,228,32,19,35,96,1341
2910 DATA 162,64,169,0,157,68,3,169,6,157,69,3,169,5,157,66,3,1427
2920 DATA 169,255,157,72,3,157,73,3,32,86,228,32,19,35,173,0,6,1500
2930 DATA 201,47,144,3,160,255,96,230,132,160,1,200,192,10,240,10,185,2266
2940 DATA 0,6,201,32,240,49,76,144,44,169,16,133,160,169,17,133,161,1750
2950 DATA 164,160,192,9,240,15,185,0,6,164,161,153,0,6,198,160,198,2011
2960 DATA 161,76,167,44,160,10,169,46,153,0,6,160,18,169,155,153,0,1647
2970 DATA 6,160,0,96,169,46,153,0,6,169,10,133,160,200,132,161,164,1765
2980 DATA 160,192,13,240,15,185,0,6,164,161,153,0,6,230,160,230,161,2076
2990 DATA 76,217,44,164,161,169,32,153,0,6,200,192,14,208,248,160,16,2060
3000 DATA 162,17,185,0,6,157,0,6,202,136,192,12,208,244,160,18,169,1874
3010 DATA 155,153,0,6,160,0,96,162,0,160,1,189,0,6,145,129,201,1563
3020 DATA 155,240,17,232,165,129,24,105,1,133,129,165,130,105,0,133,130,1993
3030 DATA 76,24,45,96,169,6,133,84,169,16,133,85,133,82,162,0,160,1573
3040 DATA 2,189,110,6,201,155,240,8,153,142,45,75,45,70,46,232,200,1919
3050 DATA 76,65,45,169,48,141,153,45,141,152,45,166,132,240,21,238,153,2030
3060 DATA 45,173,153,45,201,58,208,8,169,48,141,153,45,238,152,45,202,2084
3070 DATA 208,235,162,142,160,45,32,167,43,160,4,162,0,189,0,6,157,1872
3080 DATA 163,45,232,136,208,246,162,161,160,45,32,167,43,96,32,32,68,2028
3090 DATA 49,58,42,46,42,95,95,48,48,32,105,116,101,109,115,155,32,1288
3100 DATA 32,48,48,48,32,70,114,101,101,32,83,101,99,116,111,114,115,1365
3110 DATA 155,169,0,133,135,169,128,133,136,166,137,240,17,165,135,24,105,2147
3120 DATA 20,133,135,165,136,105,0,133,136,202,76,190,45,169,16,133,82,1876
3130 DATA 169,21,133,84,169,155,32,146,43,169,9,133,84,162,0,169,11,1689
3140 DATA 157,66,3,165,135,157,68,3,165,136,157,69,3,169,0,164,132,1749
3150 DATA 240,11,24,105,20,201,180,240,4,136,76,247,45,157,72,3,169,1930
3160 DATA 0,157,73,3,32,86,228,96,169,5,197,158,208,22,165,177,208,1984
3170 DATA 18,198,137,165,137,201,255,240,6,198,133,32,180,45,96,169,0,2210
3180 DATA 133,137,96,169,19,197,158,208,26,165,177,208,22,230,133,165,133,2376
3190 DATA 197,134,240,12,165,134,48,8,240,6,230,137,71,46,77,46,32,1823
3200 DATA 180,45,96,198,133,96,78,46,73,47,165,158,201,0,208,44,162,1930
3210 DATA 129,160,46,169,3,32,125,36,240,53,162,133,160,46,169,3,32,1698
3220 DATA 125,36,240,45,162,137,160,46,169,3,32,125,36,240,37,162,141,1896
3230 DATA 160,46,169,6,32,125,36,240,29,96,36,101,115,107,38,105,108,1549
3240 DATA 101,54,105,101,119,47,112,116,105,111,110,115,76,224,47,76,62,1681
3250 DATA 48,76,201,48,76,68,49,162,194,160,46,169,4,32,125,36,240,1734
3260 DATA 36,162,199,160,46,169,3,32,125,36,240,28,162,203,160,46,169,1976
3270 DATA 3,32,125,36,240,20,96,34,97,115,105,99,34,111,111,116,41,1415
3280 DATA 110,102,111,76,176,49,76,184,49,76,192,49,162,38,160,47,169,1826
3290 DATA 4,32,125,36,240,116,162,43,160,47,169,6,32,125,36,240,108,1681
3300 DATA 162,50,160,47,169,8,32,125,36,240,100,162,59,160,47,169,5,1731
3310 DATA 32,125,36,240,92,162,65,160,47,169,9,32,125,36,240,84,162,1816
3320 DATA 74,160,47,169,9,32,125,36,240,76,162,83,160,47,169,3,32,1624
3330 DATA 125,36,240,68,96,37,114,97,115,101,48,114,111,116,101,99,116,1734
3340 DATA 53,110,112,114,111,116,101,99,116,50,101,110,97,109,101,38,111,1649
3350 DATA 114,109,97,116,86,51,36,74,47,69,48,38,111,114,109,97,116,1432
3360 DATA 86,36,36,35,111,112,121,76,30,51,76,60,51,76,90,51,76,1174
```



```
3370 DATA 120,51,76,150,51,76,185,51,162,128,160,47,169,19,32,125,36,1638
3380 DATA 240,52,162,148,160,47,169,19,32,125,36,240,44,96,36,105,114,1825
3390 DATA 101,99,116,111,114,121,86,111,110,86,115,99,114,101,101,110,14,1709
3400 DATA 36,105,114,101,99,116,111,114,121,86,105,110,86,119,105,110,100,1738
3410 DATA 111,119,14,76,91,50,76,107,50,162,197,160,47,169,10,32,125,1596
3420 DATA 36,240,33,162,207,160,47,169,11,32,125,36,240,25,96,50,101,1770
3430 DATA 113,117,101,115,116,86,111,110,50,101,113,117,101,115,116,86,111,1779
3440 DATA 102,102,76,123,50,76,139,50,165,175,208,16,169,255,133,175,32,2046
3450 DATA 190,48,32,164,38,32,247,47,32,11,48,76,59,48,169,2,133,1376
3460 DATA 128,162,3,160,1,169,9,133,142,169,6,133,140,32,18,43,96,1544
3470 DATA 169,5,133,82,133,85,169,2,133,84,162,37,160,48,169,22,133,1726
3480 DATA 168,169,0,133,169,32,154,36,96,68,101,115,107,155,155,66,97,1821
3490 DATA 115,105,99,155,66,111,111,116,155,73,110,102,111,155,104,104,96,1888
3500 DATA 165,175,208,18,169,255,133,175,70,48,65,49,133,177,32,190,48,2110
3510 DATA 32,164,38,32,87,48,32,107,48,76,59,48,169,2,133,128,162,1365
3520 DATA 9,160,1,169,13,133,142,169,9,133,140,32,18,43,96,169,11,1447
3530 DATA 133,82,133,85,169,2,133,84,162,133,160,48,169,57,133,168,169,2020
3540 DATA 0,133,169,32,154,36,96,70,105,108,101,155,155,69,114,97,115,1709
3550 DATA 101,155,80,114,111,116,101,99,116,155,85,110,112,114,111,116,101,1897
3560 DATA 99,116,155,82,101,110,97,109,101,155,70,111,114,109,97,116,22,1764
3570 DATA 83,68,155,70,111,114,109,97,116,22,68,68,155,169,0,133,176,1714
3580 DATA 165,178,240,2,230,176,96,165,175,208,18,169,255,133,175,133,177,2695
3590 DATA 32,190,48,32,164,38,32,226,48,32,246,48,76,59,48,169,2,1490
3600 DATA 133,128,162,13,160,1,169,26,133,142,169,5,133,140,32,18,43,1607
3610 DATA 96,169,15,133,82,133,85,169,2,133,84,162,16,160,49,169,52,1709
3620 DATA 133,168,169,0,133,169,32,154,36,96,86,105,101,119,155,155,68,1879
3630 DATA 105,114,101,99,116,111,114,121,22,111,110,22,115,99,114,101,101,1676
3640 DATA 110,46,32,32,155,68,105,114,101,99,116,111,114,121,22,105,110,1561
3650 DATA 22,119,105,110,100,111,119,46,32,66,49,61,50,23,155,165,175,1508
3660 DATA 208,18,169,255,133,175,133,177,32,190,48,32,164,38,32,93,49,1946
3670 DATA 32,113,49,76,59,48,169,2,133,128,162,21,160,1,169,17,133,1472
3680 DATA 142,169,5,133,140,32,18,43,96,169,23,133,82,133,85,169,2,1574
3690 DATA 133,84,162,139,160,49,169,37,133,168,169,0,133,169,32,154,36,1927
3700 DATA 96,79,112,116,105,111,110,115,155,155,82,101,113,117,101,115,116,1899
3710 DATA 22,111,110,32,32,23,155,82,101,113,117,101,115,116,22,111,102,1465
3720 DATA 102,32,32,155,169,10,32,183,36,76,116,228,169,11,32,183,36,1602
3730 DATA 76,119,228,32,156,38,32,215,49,32,235,49,173,132,2,208,251,2027
3740 DATA 32,90,41,32,180,38,76,59,48,169,2,133,128,162,10,160,4,1364
3750 DATA 169,21,133,142,169,12,133,140,32,18,43,96,169,12,133,82,133,1637
3760 DATA 85,169,5,133,84,162,5,160,50,169,86,133,168,169,0,133,169,1880
3770 DATA 32,154,36,96,73,110,102,111,155,155,84,79,83,44,155,155,86,1710
3780 DATA 101,114,115,105,111,110,32,49,46,48,155,98,121,32,77,97,114,1525
3790 DATA 99,32,69,98,110,101,114,155,40,99,41,32,50,54,46,49,50,1239
3800 DATA 46,49,57,56,53,155,155,155,112,108,62,50,57,51,101,97,115,1479
3810 DATA 101,32,99,108,105,99,107,32,111,110,99,101,155,116,111,32,99,1617
3820 DATA 111,110,116,105,110,117,101,46,155,169,32,141,66,49,169,23,141,1761
3830 DATA 43,49,32,222,43,76,59,48,169,32,141,43,49,169,23,141,66,1405
3840 DATA 49,32,222,43,76,59,48,169,32,141,174,49,169,23,141,160,49,1636
3850 DATA 32,222,43,76,59,48,169,32,141,160,49,169,23,141,174,49,32,1619
3860 DATA 222,43,76,59,48,133,174,169,0,141,240,2,141,29,208,32,156,1873
3870 DATA 38,32,189,50,32,209,50,32,235,50,169,3,141,240,2,141,29,1642
3880 DATA 208,32,180,38,96,169,0,133,128,162,2,160,13,169,36,133,142,1801
3890 DATA 169,5,133,140,32,18,43,96,169,4,133,82,133,85,169,15,133,1559
3900 DATA 84,165,174,32,82,37,169,155,32,146,43,169,58,32,146,43,96,1663
3910 DATA 162,0,169,5,157,66,3,169,28,157,68,3,169,6,157,69,3,1391
3920 DATA 169,255,157,72,3,157,73,3,32,86,228,32,19,35,162,0,189,1672
3930 DATA 28,6,201,32,240,4,232,76,12,51,232,169,155,157,28,6,96,1725
3940 DATA 169,102,32,155,50,173,28,6,201,32,240,15,162,33,160,0,32,1590
3950 DATA 60,33,169,16,32,99,39,32,82,42,76,58,51,219,51,59,48,1166
3960 DATA 169,103,32,155,50,173,28,6,201,32,240,15,162,35,160,0,32,1593
3970 DATA 60,33,169,16,32,99,39,32,82,42,76,59,48,169,104,32,155,1247
3980 DATA 50,173,28,6,201,32,240,15,162,36,160,0,32,60,33,169,16,1413
3990 DATA 32,99,39,32,82,42,76,59,48,169,106,32,155,50,173,28,6,1228
4000 DATA 201,32,240,15,162,32,160,0,32,60,33,169,16,32,99,39,32,1354
4010 DATA 82,42,76,59,48,169,100,32,155,50,173,28,6,201,32,240,20,1513
4020 DATA 169,0,32,183,36,162,253,160,0,32,60,33,169,16,32,99,39,1475
```

```
4030 DATA 32,82,42,76,59,48,169,108,32,155,50,173,28,6,201,32,240,1533
4040 DATA 20,169,8,32,183,36,162,254,160,0,32,60,33,162,16,32,99,1458
4050 DATA 39,32,82,42,76,59,48,0,96,251,96,0,0,0,0,0,821
4060 DATA 0,0,0,24,24,24,24,0,24,0,0,102,102,102,0,0,0,426
4070 DATA 0,0,102,255,102,102,255,102,0,24,62,96,60,6,124,24,0,1314
4080 DATA 0,102,108,24,48,102,70,0,28,54,28,56,111,102,59,0,0,892
4090 DATA 24,24,24,0,0,0,0,0,14,28,24,24,28,14,0,0,112,316
4100 DATA 56,24,24,56,112,0,0,102,60,255,60,102,0,0,0,24,24,899
4110 DATA 126,24,24,0,0,0,0,0,0,0,24,24,48,0,0,0,126,396
4120 DATA 0,0,0,0,0,0,0,0,24,24,0,0,6,12,24,48,138
4130 DATA 96,64,0,0,60,102,110,118,102,60,0,0,24,56,24,24,24,864
4140 DATA 126,0,0,60,102,12,24,48,126,0,0,126,12,24,12,102,60,834
4150 DATA 0,0,12,28,60,108,126,12,0,0,126,96,124,6,102,60,0,860
4160 DATA 0,60,96,124,102,102,60,0,0,126,6,12,24,48,48,0,0,808
4170 DATA 60,102,60,102,102,60,0,0,60,102,62,6,12,56,0,0,0,784
4180 DATA 24,24,0,24,24,0,0,0,24,24,0,24,24,48,6,12,24,282
4190 DATA 48,24,12,6,0,0,0,126,0,0,126,0,0,96,48,24,12,522
4200 DATA 24,48,96,0,0,60,102,12,252,96,247,97,24,0,24,0,0,1082
4210 DATA 60,102,110,110,96,62,0,0,24,60,102,102,126,102,0,0,124,1180
4220 DATA 102,124,102,102,124,0,0,60,102,96,96,102,60,0,0,120,108,1298
4230 DATA 102,102,108,120,0,0,126,96,124,96,96,126,0,0,126,96,124,1442
4240 DATA 96,96,96,0,0,62,96,96,110,102,62,0,0,102,102,126,102,1248
4250 DATA 102,102,0,0,126,24,24,24,24,126,0,0,6,6,6,6,102,678
4260 DATA 60,0,0,102,108,120,120,108,102,0,0,96,96,96,96,96,126,1326
4270 DATA 0,0,99,119,127,107,99,99,0,0,102,118,126,126,110,102,0,1334
4280 DATA 0,60,102,102,102,102,60,0,0,124,102,102,124,96,96,0,0,1172
4290 DATA 60,102,102,102,108,54,0,0,124,102,102,124,108,102,0,0,60,1250
4300 DATA 96,60,6,6,60,0,0,126,24,24,24,24,24,0,0,102,102,678
4310 DATA 102,102,102,126,0,0,102,102,102,102,60,24,0,0,99,99,107,1229
4320 DATA 127,119,99,0,0,102,102,60,60,102,102,0,0,102,102,60,24,1161
4330 DATA 24,24,0,0,126,12,24,48,96,126,0,0,30,24,24,24,24,606
4340 DATA 30,0,0,64,96,48,24,12,6,0,0,120,24,24,24,24,120,616
4350 DATA 0,0,8,28,54,99,0,0,0,248,97,243,98,0,0,0,0,875
4360 DATA 0,0,255,0,0,0,0,0,0,0,0,255,128,190,162,190,1180
4370 DATA 128,129,130,255,1,1,1,1,2,129,65,130,129,128,129,129,129,1616
4380 DATA 128,255,65,129,1,129,129,129,1,255,1,62,194,192,127,96,100,1993
4390 DATA 118,192,62,35,3,254,6,38,110,118,116,115,117,118,118,118,118,1756
4400 DATA 110,14,238,206,46,110,110,110,118,118,118,118,118,56,63,0,110,1763
4410 DATA 110,110,110,110,28,248,0,255,129,189,189,189,189,129,255,255,129,2624
4420 DATA 153,189,153,153,129,255,255,129,129,129,129,129,129,255,255,129,153,28
53
4430 DATA 153,189,153,129,255,129,129,129,129,129,129,129,129,255,0,0,0,2166
4440 DATA 0,0,0,255,0,255,0,0,0,0,0,255,192,224,176,152,1509
4450 DATA 140,134,131,255,3,7,13,25,49,97,193,131,134,140,152,176,224,2004
4460 DATA 192,255,193,97,49,25,13,7,3,255,0,0,0,0,0,0,1089
4470 DATA 0,3,2,6,4,12,200,120,48,0,0,0,0,0,0,0,395
4480 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,120,120
4490 DATA 96,120,96,126,24,30,0,0,24,60,126,24,24,24,0,0,24,798
4500 DATA 24,24,126,60,24,0,0,24,48,126,244,98,239,99,48,24,0,1208
4510 DATA 0,0,24,12,126,12,24,0,0,0,24,60,126,126,60,24,0,618
4520 DATA 0,0,60,6,62,102,62,0,0,96,96,124,102,102,124,0,0,936
4530 DATA 0,60,96,96,96,60,0,0,6,6,62,102,102,62,0,0,0,748
4540 DATA 60,102,126,96,60,0,0,14,24,62,24,24,24,0,0,0,62,678
4550 DATA 102,102,62,6,124,0,96,96,124,102,102,102,0,0,24,0,56,1098
4560 DATA 24,24,60,0,0,6,0,6,6,6,6,60,0,96,96,108,120,618
4570 DATA 108,102,0,0,56,24,24,24,24,60,0,0,0,102,127,127,107,885
4580 DATA 99,0,0,0,124,102,102,102,102,0,0,0,60,102,102,102,60,1057
4590 DATA 0,0,0,124,102,102,124,96,96,0,0,62,102,102,62,6,6,984
4600 DATA 0,0,124,102,96,96,96,0,0,0,62,96,60,6,124,0,0,862
4610 DATA 24,126,24,24,24,14,0,0,0,102,102,102,102,62,0,0,0,706
4620 DATA 102,102,102,60,24,0,0,0,99,107,127,62,54,0,0,0,102,941
4630 DATA 60,24,60,102,0,0,0,102,102,102,62,12,120,0,0,126,12,884
4640 DATA 24,48,126,0,0,24,60,126,126,24,60,0,24,24,24,24,24,738
4650 DATA 24,24,24,0,126,120,124,110,102,6,0,240,99,255,99,8,24,1385
4660 DATA 56,120,56,24,8,0,16,24,28,30,28,24,16,0,224,2,225,881
4670 DATA 2,0,32,34
```

GEM-Routinen

Dieses Programm enthält ab Zeile 30000 drei nützliche Unterprogramme, die von den GEM-Fähigkeiten des STs regen Gebrauch machen. Die Zeilen 1000-1210 sind nur Demonstrationsbeispiele. Zu bedenken ist, daß das Unterprogramm FARBWAHL natürlich nur mit Farbmonitor sinnvoll zu gebrauchen ist und daß der Wert der Variablen FAR im Unterprogramm PIE bei einem SW-Monitor immer 1 sein muß. Nachfolgend die genauen Beschreibungen der drei Routinen.

Farbwahl

Mit diesem Unterprogramm können die Farben der 16 Farbregister verändert werden, was sonst nur im Kontrollfeld möglich ist. Vor dem Aufruf müssen folgende Variablen mit den gewünschten Werten versorgt werden:

REGISTER – Nummer des Farbregisters (0-15)
ROT – Intensität des Rot-Anteils (0-999)
GRUEN – Intensität des Grün-Anteils (0-999)
BLAU – Intensität des Blau-Anteils (0-999)

Textausgabe

Mit diesem Unterprogramm können Sie einen Text, den Sie in TEXT$ übergeben haben, sehr komfortabel in verschiedenen Größen und Textarten ausgeben. Hier die Variablen, die vor dem Aufruf bestimmt werden müssen:

GROESSE – Höhe der Zeichen in Pixel; Normalwert 9 (Farbe), 17 (SW).

WINKEL – Hiermit kann die Basislinie des Textes verändert werden; der Text wird also gedreht. Möglich sind allerdings nur folgende Werte:

0: keine Veränderung
900: Text um 90° n. links
1800: Text auf dem Kopf
2700: Text um 90° n. rechts

Hierbei ist zu beachten, daß sich auch die Schreibrichtung ändert.

EFFECT – Durch einen Wert ungleich 0 kann der Text in verschiedenen Schriftarten dargestellt werden.

1: fette Schrift
2: helle Schrift
4: Kursivschrift
8: Text wird unterstrichen
16: Text wird umrahmt
32: Text wird schattiert

Um nun mehrere Effekte zu erreichen, müssen Sie nur die Werte addieren. Z.B. EFFECT=9 (fette Schrift unterstrichen)

XTEXT – X-Koordinate des Textbeginns.
YTEXT – Y-Koordinate des Textbeginns.

Pie

Hiermit können ausgefüllte Ellipsen-Ausschnitte erzeugt werden, was zum Beispiel für Kuchen-Dia-

16 Bit

gramme sehr nützlich ist. Als Variablen müssen bestimmt werden:

FAR – Nummer des Farbregisters, dessen Farbe erscheinen soll.

FUELLTYP
0: Ellipse wird nicht ausgefüllt.
1: Ellipse wird komplett mit Farbe gefüllt.
2: Ellipse wird punktiert dargestellt.
3: Ellipse wird schraffiert dargestellt.

X – X-Koordinate der Ellipsen-Mitte.
Y – Y-Koordinate der Ellipsen-Mitte.
XRAD – Radius in X-Richtung.
YRAD – Radius in Y-Richtung.
SWIN – Startwinkel des Ausschnittes (0-3600).
EWIN – Endwinkel des Ausschnittes (0-3600).

Dieses Unterprogramm ist sehr vielseitig. Damit können z.B. Halbkreise und Sinuskurven erstellt werden.

Christian Rduch

Listing

```
   1000  fullw 2:clearw 2
   1010  ?"FARBWAHL-Demo"
   1020  register=0:rot=999:gruen=999:bl
au =0:gosub farbwahl
   1030  for i = 1 to 9999:next
   1040  register=0:rot=699:gruen=399:bl
au =500:gosub farbwahl
   1050  for i = 1 to 9999:next
   1060  fullw 2:clearw 2
   1070  ?"TEXTAUSGABE-Demo"
   1080  text$="RADDISH-SOFT gr"+chr$(&H
81)+chr$(&HE1)+"t"
   1090  groesse=16:winkel=0:effect=16:x
text=20:ytext=100
   1093  gosub textausgabe
   1095  text$="Computer Kontakt  "
   1097  groesse=16:winkel=0:effect=16:x
text=60:ytext=130
   1098  gosub textausgabe
   1100  groesse=7:winkel=1800:effect=0:
xtext=200:ytext=150
   1'01  gosub textausgabe:text$="RADDIS
H-SOFT gr"+chr$(&H81)+chr$(&HE1)+"t"
   1102  groesse=7:winkel=1800:effect=0:
xtext=220:ytext=160
   1105  gosub textausgabe
   1106  text$="":groesse=9:winkel=0:gos
ub textausgabe
   1110  for i = 1 to 9999:next
   1120  fullw 2:clearw 2
   1130  ?"PIE-Demo"
   1140  far=1:fuelltyp=1:x=100:y=100
   1150  xrad=40:yrad=20:swin=0:ewin=320
0
   1160  gosub pie
   1170  fuelltyp=2:x=230:y=100:xrad=70
   1180  yrad=70:swin=1400:ewin=2500
   1190  gosub pie
   1200  for i = 1 to 9999:next
   1210  end
   30000 textausgabe:
   30001 poke contrl,107:poke contrl+2,0
:poke contrl+6,1:poke intin,groesse:vdis
ys
   30002 poke contrl,13:poke contrl+2,0:
poke contrl+6,1:poke intin,winkel:vdisys
   30003 poke contrl,106:poke contrl+2,0
:poke contrl+6,1:poke intin,effect:vdisy
s
   30004 poke contrl,8:poke contrl+2,1:p
okE contrl+6,LEN(TEXT$)
   30005 FOR QW = 0 TO LEN(TEXT$)-1:POKE
 INTIN+QW*2,ASC(MID$(TEXT$,QW+1,1)):NEXT
   30006 poke ptsin,xtext:poke ptsin+2,y
teXt:vdisys
   30007 return
   30100 farbwahl:
   30101 poke contrl,14:poke contrl+2,0:
poke contrl+6,4:poke intin,register
   30102 poke intin+2,rot:poke intin+4,g
ruen:poke intin+6,blau:vdisys
   30103 return
   30200 pie:
   30201 poke contrl,25:poke contrl+2,0:
poke contrl+6,1:poke intin,far  :vdisys
   30202 poke contrl,23:poke contrl+2,0:
poke contrl+6,1:poke intin,fuelltyp:vdis
ys
   30203 poke contrl,11:poke contrl+2,2:
poke contrl+6,2:poke contrl+10,7
   30204 poke ptsin,x:poke ptsin+2,y:pok
e ptsin+4,xrad:poke ptsin+6,yrad
   30205 poke intin,swin:poke intin+2,ew
in:vdisys
   30206 return
```


Nützliche Routinen für ST-Assembler-Programmierer

Der Atari ST mit seinem Superprozessor 68000 läßt sich ausgezeichnet in Assembler programmieren. Es ist allerdings lästig, Ein-, Ausgaberoutinen usw. jedesmal neu zu schreiben oder in das Textfile einzukopieren. Auch die Fehlersuche während der Entwicklungsphase eines Assembler-Programms gestaltet sich manchmal nicht ganz einfach, wenn es auch leistungsfähige Werkzeuge wie SID gibt.

Im folgenden lernen wir ein Hilfsprogramm kennen, das eine Reihe von nützlichen Routinen zur Verfügung stellt. Es übernimmt die notwendigen Schritte beim Programmstart, wie das Bereitstellen eines Speicherbereichs für den Stack, und offeriert Routinen zur Ein- und Ausgabe von Strings und Zahlen sowie zur Unterstützung bei der Fehlersuche. Letztere stellen beispielsweise einen Speicherausgang dar oder zeigen die aktuelle Belegung der Register an.

Sicher gibt es unter den Lesern etliche, die noch nie ein Assembler-Programm für den ST geschrieben haben. Daher soll hier einmal detailliert erklärt werden, wie man zweckmäßig vorgeht. Wer das Gebiet bereits beherrscht, braucht nicht weiterzulesen!

ST-Entwicklungspaket

Wir wollen davon ausgehen, daß die Standardprogramme des ST-Entwicklungspakets verwendet werden. Im einzelnen braucht man folgende Programme, die auf der benutzten Diskette bzw. in der RAM-Floppy vorhanden sein müssen:

BATCH.TTP
AS68.PRG
LINK68.PRG
RELMOD.PRG
RM.PRG
WAIT.PRG

Außerdem ist selbstverständlich ein Texteditor nötig. Mit seiner Hilfe wird das Assembler-Listing eingegeben; das File bezeichnen wir als HELP.S. Man sollte sich angewöhnen, alle Dateien, die einen Assembler-Sourcetext enthalten, mit der Endung (Extension) .S zu versehen. Durch die systematische Vergabe von Extensions (die ja schon damit beginnt, daß .ACC, .PRG, .TOS usw. ganz bestimmte Bedeutungen für das ST-Betriebssystem haben) bringt man eine gewisse Ordnung und Übersichtlichkeit in seine Dateiensammlung.

Wir erzeugen ein weiteres File, bezeichnet als ASSO.BAT, mit folgendem kurzen Inhalt:

as68 −1 %1.s
wait

Jetzt läßt sich der Assembler auf das Sourcefile ansetzen, indem BATCH.TTP angeklickt und als Parameterzeile getippt wird: asso help

Groß- und Kleinschreibung spielen keine Rolle. Wichtig ist, daß man grundsätzlich nur ein einziges Blank zur Trennung von Wörtern einsetzt, weil einige Programme, so auch BATCH.TTP, die Eingabe sonst gründlich mißverstehen.

Wenn wir alles richtig gemacht haben, finden wir in unserer Library nun ein neues File, HELP.0. Dieses enthält den übersetzten Sourcecode, allerdings noch nicht in seiner endgültigen, ausführbaren Form. Vielmehr sind noch Informationen über die im Sourcetext auftretenden Labels enthalten, soweit sie mittels der

16 Bit

Assembler-Anweisung (Direktive) .globl für global, also auch für andere Programme verfügbar, erklärt wurden. Durch einen Linker (= Verbinder) lassen sich mehrere .0-Files zu einem einzigen, ausführbaren File zusammenfügen. Diese Aufgabe übernehmen LINK 68 und RELMOD.

Testprogramm

Jeder Programmteil kann auch diejenigen Labels aus anderen Teilen benutzen (etwa in Form eines Unterprogrammaufrufs), die als global definiert sind. In unserem konkreten Fall bedeutet dies, daß wir die Routinen des Help-Programms aufrufen können. Wie das geht, sehen wir schnell, wenn wir als TEST.S das abgedruckte Testprogramm eingeben.

Um dieses, wie auch weitere, zu übersetzen, benutzen wir am besten das ebenfalls abgebildete und einzutippende File ASS.BAT: einfach BATCH.TTP aufrufen und ass test eingeben. So wird das ausführbare File TEST.TOS erzeugt. Nach diesem Schema, also unter Verwendung von ASS.BAT, lassen sich Assembler-Programme schnell übersetzen.

Das Sourcefile HELP.S ist übrigens nicht mehr erforderlich; um Platz auf der Assembler-Diskette zu sparen, kann es auf eine andere Diskette ausgelagert werden. Gleiches gilt für ASSO.BAT, das ja nur benötigt wird, um .0-Files zu generieren.

Michael Schramm

HELP.S

```
* Initialisierung und wichtige
* Systemaufrufe sowie Debugging-
* hilfen für Assemblerprogramme

.globl _main,_error,_exit
.globl _testkey,_conin,_writeln
.globl _newline,_write,_beep
.globl _blank,_conout,_readln
.globl _marke,_hexprnt,_hexreg
.globl _decprnt,_reglist
.globl _getnumb,_hexdump

CR = $0d
LF = $0a

        .text
* Nicht benötigten Speicherplatz
* ans Betriebssystem zurückgeben
* und den Stack vorbereiten.

start       move.l 4(sp),a5
            lea eost+2,sp
            move.l $C(a5),d0
            add.l $14(a5),d0
            add.l $1C(a5),d0
            add.l #$100,d0
            move.l d0,-(sp)
            move.l a5,-(sp)
            clr.w -(sp)
            move.w #$4A,-(sp)
            trap #1
            add.l #12,sp
            tst.l d0
            beq _main

* das Hauptprogramm muß mit
* dem Label _main beginnen.

_error      lea errorstr(pc),a0
            bsr _writeln

* bra _exit  beendet das Prgm.

_exit       lea termstr(pc),a0
            bsr _write
exitwait    bsr _testkey
            bne exitwait
            bsr _conin
            clr.w -(sp)
            trap #1

* Prüfen, ob Taste gedrückt ist.
* Tastencode in D0.W liefern
* ( = 0, falls keine Taste).

* Z-Flag setzen, f. keine Taste.

_testkey    move.l a0,-(sp)
            move.l #$000600ff,-(sp)
            trap #1
            addq.l #4,sp
            movea.l (sp)+,a0
            tst.w d0
            rts

* Auf Tastendruck warten,
* Tastencode in D0.W. Es
* wird nichts gedruckt.

_conin      bsr _testkey
            beq _conin
            rts

* String drucken, der ab A0
* abgelegt ist. Bei _writeln
* Cursor nach der Ausgabe
* auf nächsten Zeilenanfang.

_writeln    bsr _write
_newline    movem.l d0-d4/a0-a6,-(sp)
            lea newline(pc),a0
            bra write1
_write      movem.l d0-d4/a0-a6,-(sp)
write1      move.l a0,-(sp)
            move.w #9,-(sp)
            trap #1
            addq.l #6,sp
            bra wr_rstr

* Akustisches Signal.

_beep       movem.l d0-d4/a0-a6,-(sp)
            moveq #7,d0
            bra conout1

* Blank drucken.

_blank      movem.l d0-d4/a0-a6,-(sp)
            moveq #32,d0
            bra conout1

* D0.B als Character ausgeben.

_conout     movem.l d0-d4/a0-a6,-(sp)
conout1     move.w d0,-(sp)
            move.w #2,-(sp)
            trap #1
            addq.l #4,sp
wr_rstr     movem.l (sp)+,d0-d4/a0-a6
            rts

* Eingabe eines maximal D0.B
* langen Strings. Als Ergebnis
*  Eingabelänge in D0, A0
* Zeiger auf den String.

_readln     lea strngbuf,a0
            move.l a0,-(sp)
            move.b d0,(a0)
            move.w #10,-(sp)
            trap #1
            addq.l #6,sp
            lea strngbuf+1,a0
            clr.l d0
            move.b (a0)+,d0
            clr.b 0(a0,d0.w)
            bra _newline

* Beim Testen von Programmen, um
* festzustellen, ob bestimmte
* Stellen erreicht werden:
*    bsr _marke
*    .dc.w nummer
* nummer wird mit ausgegeben.

_marke      movem.l d0-d4/a0-a6,-(sp)
            move sr,-(sp)
            lea marktext(pc),a0
            bsr _write
            movea.l 50(sp),a0
            clr.l d0
            move.w (a0)+,d0
            move.l a0,50(sp)
            bsr _decprnt
            bsr _newline
            move (sp)+,ccr
            bra wr_rstr

* D0.B hexadezimal drucken.

_hexprnt    rol.b #4,d0
            bsr hexdigit
            rol.b #4,d0
hexdigit    move.l d1,-(sp)
            move.b d0,d1
            andi.b #$f,d1
            addi.b #'0',d1
            cmpi.b #'9'+1,d1
            bcs hex_ziff
            addi.b #'A'-'0'-10,d1
hex_ziff    exg d1,d0
            bsr _conout
            exg d1,d0
            move.l (sp)+,d1
            rts
```

```
* D0.L hexadezimal drucken.
_hexreg   move.l d1,-(sp)
          moveq #3,d1
reg_hlp   rol.l #8,d0
          bsr _hexprnt
          dbf d1,reg_hlp
          move.l (sp)+,d1
          rts
* D0.L dezimal drucken.
_decprnt  movem.l d0-d4/a0-a6,-(sp)
          bclr #7,d4
          moveq #36,d3
          lea pot10tab(pc),a0
dezpr_nx  move.l 0(a0,d3.w),d1
          moveq #-1,d2
dezpr_lp  addq.b #1,d2
          sub.l d1,d0
          bcc dezpr_lp
          add.l d1,d0
          tst.b d2
          beq dez_pr_0
          bset #7,d4
dezpr_dg  addi.b #'0',d2
dezpr_bt  exg d2,d0
          bsr _conout
          exg d2,d0
          subq.w #4,d3
          bcc dezpr_nx
wr_rstr1  bra wr_rstr
dez_pr_0  tst.b d4
          bmi dezpr_dg
          subq.b #1,d1
          beq dezpr_dg
          moveq #32,d2
          bra dezpr_bt
* Alle Registerinhalte drucken.
_reglist  movem.l a0-a6/d0-d4,-(sp)
          move sr,-(sp)
          movem.l d0-d7/a0-a6,-(sp)
          bsr _newline
          moveq #$30,d3  * '0'
reg_loop  bsr _blank
          moveq #$44,d0  * 'D'
          move.l (sp)+,d1
          bsr reg_out
          cmpi.b #'7',d3
          bcc reg_pc
          moveq #$41,d0  * 'A'
          move.l 28(sp),d1
          bsr reg_out
          bsr _newline
          addq.w #1,d3
          bra reg_loop
reg_pc    adda.w #28,sp
          lea flagname(pc),a0
          bsr _writeln
          lea pc_text(pc),a0
          bsr _write
          move.l 50(sp),d1
          bsr reg_outl
          lea sr_text(pc),a0
          bsr _write
          move.w (sp),d1
          bsr reg_bit1
          bsr reg_bit1
          bsr reg_bit1
          moveq #1,d2
          bsr reg_bit
          moveq #2,d2
          bsr reg_bit
          bsr _blank
          moveq #2,d2
          bsr reg_bit
          moveq #4,d2
          bsr reg_bit
          bsr _newline
restore   move (sp)+,ccr
wr_rstr2  bra wr_rstrl
reg_out   bsr _conout
          move.l d3,d0
          bsr _conout
reg_outl  lea equtext(pc),a0
          bsr _write
          move.l d1,d0
          bsr _decprnt
          lea equtext(pc),a0
          bsr _write
          bsr _hexreg
          moveq #9,d0
blank_lp  bsr _blank
          dbf d0,blank_lp
          rts
reg_bit1  clr.w d2
reg_bit   bsr _blank
regb_lp   moveq #$30,d0
          lsl.w #1,d1
          bcc regb_bit
          addq.w #1,d0
regb_bit  bsr _conout
          dbf d2,regb_lp
          rts
* Liest eine Dezimalzahl ab (A0)+
* und liefert ihr binäres
* Äquivalent in D0. Der String
* muß mit einem Nullbyte enden.
* Falls der String keine Zahl
* darstellt oder sich die Zahl
* als größer als 2**32-1
```



```
* erweist, wird das Lesen
* abgebrochen und das C-Flag
* gesetzt.
_getnumb  movem.l d1-d3,-(sp)
          clr.l d0
          clr.l d1
getn_lp   move.b (a0)+,d1
          beq getn_rst
          subi.b #'0',d1
          bcs getn_rst
          cmpi.b #10,d1
          bcc getn_err
getn_dig  lsl.l #1,d0
          bcs getn_rst
          move.l d0,d3
          lsl.l #1,d0
          bcs getn_rst
          lsl.l #1,d0
          bcs getn_rst
          add.l d3,d0
          bcs getn_rst
          add.l d1,d0
          bcc getn_lp
getn_err  move #1,ccr    * C-Flag!
getn_rst  movem.l (sp)+,d1-d3
          rts
* Hexdump von D0.W Bytes
* ab Adresse A0
hexdump   movem.l d0-d4/a0-a6,-(sp)
          move sr,-(sp)
          bsr _newline
          movea.l a0,a1
          move.w d0,d1
          lea dumptext(pc),a0
          bsr _write
          move.l a1,d0
          bsr _decprnt
          bsr _newline
dumppage  moveq #19,d2
dumpline  bsr _blank
          move.l a1,d0
          bsr _hexreg
          bsr _blank
          moveq #1,d3
dumphex1  moveq #7,d4
          bsr _blank
dumphex2  move.b (a1)+,d0
          bsr _hexprnt
          dbf d4,dumphex2
          dbf d3,dumphex1
          bsr _blank
          suba.w #16,a1
          moveq #1,d3
dumpasc1  moveq #7,d4
          bsr _blank
dumpasc2  move.b (a1)+,d0
          cmpi.b #32,d0
          bcc dumpasc3
          moveq #$2e,d0  * '.'
dumpasc3  bsr _conout
          dbf d4,dumpasc2
          dbf d3,dumpasc1
          bsr _newline
          subi.w #16,d1
          bls dump_end
          dbf d2,dumpline
          bsr _conin
          cmpi.b #3,d0
          bne dumppage
dump_end  bra restore
pot10tab  .dc.l 1,10,100,1000,10000
          .dc.l 100000,1000000
          .dc.l 10000000,100000000
          .dc.l 1000000000
newline   .dc.b CR,LF,0
termstr   .dc.b CR,LF
          .dc.b ' Programmende, '
          .dc.b 'Taste drücken!',0
errorstr  .dc.b CR,LF
          .dc.b ' Programmabbruch'
          .dc.b ' wegen Fehler'
          .dc.b 'bedingung!',0
marktext  .dc.b ' Testpunkt ',0
equtext   .dc.b ' = ',0
flagname  .dc.b '        T  S'
          .dc.b '       I210'
          .dc.b '        XNZVC',0
pc_text   .dc.b ' PC',0
sr_text   .dc.b 'SR =',0
dumptext  .dc.b 'Hexdump ab'
          .dc.b ' Adresse',0
          .bss
          .ds.w 511
eost      .ds.w 1
strngbuf  .ds.b 260
          .end
```

TEST.S

```
as68 -l -u %1.s
link68 %1.68k=help,%1
rm %1.o
relmod %1.68k %1.tos
rm %1.68k
wait
```

ASS.BAT

```
.globl _main
.text
_main   bsr _reglist
        bsr _beep
        bra _exit
.end
```

ST-Dateiinfo-Programm

Mit Hilfe des Programms "Info" kann man sich einen schnellen Überblick über bestimmte bzw. alle Dateien einer Directory verschaffen. Man erfährt Erstelldatum, Größe, Anzahl der Wörter und Zeilen sowie kleinsten und größten Byte-Inhalt der Dateien, bekommt also wesentlich mehr Informationen, als sie das Desktop bietet. Die Ausgabe kann auch auf einem Drucker erfolgen (s. Beispielausdruck).

Das Utility ist in Assembler geschrieben, damit auch sehr große Files blitzschnell analysiert werden. Zugleich zeigt dieses Beispiel, daß sich der Atari ST sehr bequem in Assembler programmieren läßt. Der 68000 verfügt über einen leistungsfähigen, übersichtlichen Befehlssatz, und das ST-Betriebssystem (GEMDOS und BIOS) bietet effiziente Leistungen, insbesondere auch zur Dateibearbeitung.

Programmanwendung

Das assemblierte Programm wird INFO.TTP genannt. Die Extension TTP führt dazu, daß bei Aufruf vom Desktop automatisch Parameter angefordert werden. Man gibt in der Parameterzeile die Namen der zu untersuchenden Dateien durch Blanks getrennt an. Auch Sternchen und Fragezeichen sind erlaubt. Soll die Ausgabe (auch) auf einem Drucker erfolgen, so läßt man die Parameterzeile mit −P beginnen. Entsprechend steht −M für den Modem-Port (serieller Drucker). Der später folgende Ausdruck ergab sich nach Eingabe der Parameterzeile "−P *.PAS *.S *.TTP". Auf Groß- oder Kleinschreibung kommt es nicht an.

Basepage und Systemaufrufe

Wer sich für die ST-Assembler-Programmierung interessiert, sich bisher aber noch nicht an dieses Thema herangewagt hat, wird das Assemblerlisting bestimmt begrüßen. Alle Systemaufrufe (TRAPs) sind kommentiert, so daß ihre Wirkung deutlich werden dürfte. Daher möchte ich hier nur noch einige ergänzende Informationen geben.

In welcher Situation befindet man sich beim Start eines Assemblerprogramms (Startpunkt ist immer das erste Kommando im Text-Segment)? In einer etwas unsicheren, könnte man sagen, denn der Stackpointer (SP = A7) zeigt auf einen Speicherbereich, den man nicht als Stack nutzen kann. Bevor man also munter drauflosprogrammiert, sollte man den SP an das Ende eines für den Stack reservierten RAM-Bereichs setzen. Zuvor rettet man das Longword, das sich hinter der Adresse 4 (SP) verbirgt, in ein Adreßregister. Dieses Longword ist die Adresse der sogenannten Basepage. Wie der Name schon andeutet, handelt es sich hierbei um eine Speicherseite, also um einen Block von 256 Bytes Länge. Sie enthält unter anderem eine Reihe von Longwords. Die Bedeutung der wichtigsten ist im folgenden mit den zugehörigen Adressen wiedergegeben (A = Adresse der Basepage):

A+0 : A selbst
A+4 : Endadresse des Speicherbereichs, der benutzt werden darf; bei dieser Adresse beginnt der Bildspeicher, eine RAM-Floppy oder ähnliches.
A+8 : Anfangsadresse des Textsegments
A+12 : Länge des Textsegments
A+16 : Anfangsadresse des Datensegments
A+20 : Länge des Datensegments
A+24 : Anfangsadresse des BSS-Segments
A+28 : Länge des BSS-Segments
A+32 : Adresse des Parameterstrings
A+44 : Adresse, an der Informationen über den aktuellen Directory-Zugriffspfad zu finden sind.

Die Parameterstringadresse lautet normalerweise A+128; der String wird in der Basepage untergebracht und kann prinzipiell die restlichen 128 Byte ausfüllen. So lange Eingaben läßt der TTP-Mechanismus aber nicht zu. Das erste Byte des Strings gibt dessen Länge an; der eigentliche Text beginnt im nächsten Byte. Das Byte für die Längenangabe und das Stringabschlußbyte (CR = $0D) werden bei der Längenbestimmung nicht mitgezählt. Lautet die Länge Null, ist nicht einmal das CR vorhanden! Dieser Fall muß daher oftmals gesondert behandelt werden. Alle vom Benutzer eingetippten Kleinbuchstaben wurden bereits in Großbuchstaben umgewandelt. Das erste Blank im Text ist durch ein Nullbyte ersetzt; ansonsten bleibt er unverändert.

Die Information über die Länge der einzelnen Segmente ist für den Programmierer durchaus von Bedeutung, denn zu Beginn steht der gesamte freie Speicherplatz zur Verfügung. Den nicht benötigten Speicherraum sollte man dem Betriebssystem zurückgeben, indem man nur den von Basepage und Programm belegten Bereich für sich reserviert. Diese Aufgabe erledigt die Systemfunktion SETBLCK. Bei Bedarf kann man später mit MALLOC weiteren Speicherplatz anfordern oder erfragen, wie viele Bytes überhaupt noch frei sind.

Beispielausdruck

```
FLUSSDIA.PAS   30.07.86  21:15   5746   483   287   10 <= C <= 125 ASCII
PLOTTER.PAS    30.07.86  21:16   2580   258   145   10 <= C <= 122 ASCII
TERM.PAS       30.07.86  21:16   2004   265   115   10 <= C <= 122 ASCII
INFO.S         30.07.86  20:53   8696   815   369   10 <= C <= 132 ASCII
DATUM.S        30.07.86  21:16   6814   815   144   10 <= C <= 150 ASCII
PRINT.S        30.07.86  21:16   2575   276    69   10 <= C <= 150 ASCII
BATCH.TTP      30.07.86  20:40   1991   130    22    0 <= C <= 255
INFO.TTP       30.07.86  20:54   1161    51     0    0 <= C <= 255
```

Die Leistungsfähigkeit eines Betriebssystems wird im wesentlichen durch die zur Dateiver- und -bearbeitung existierenden Funktionen bestimmt. GEMDOS

16 Bit

stellt etliche Systemdienste, die in diese Kategorie fallen. Einige davon benutzt das Programm "Info". Besonders interessant sind SFIRST und SNEXT. SFIRST sucht nach einem File, dessen Name zu einem gegebenen Muster paßt. Sofern das Muster keine Sternchen oder Fragezeichen enthält, muß der Name mit dem Muster identisch sein. Andernfalls passen mehrere Dateinamen zum Muster, von denen SFIRST dann den ersten findet. SNEXT liefert sukzessive die übrigen Namen. Man erhält Name, Erstelldatum und Länge des jeweils gefundenen Files in einem 44 Byte langen Buffer, dessen Adresse dem Betriebssystem zuvor durch SETDTA mitzuteilen ist. Um an den Inhalt einer Datei heranzukommen, muß man sie mit OPEN öffnen. READ liest beliebig viele Bytes aus der Datei in einem Rutsch und schreibt sie ins RAM. Es ist zweckmäßig, immer ein Vielfaches von 512 Byte auf einmal zu lesen, weil ein Diskettensektor gerade 512 Byte aufnimmt. Wenn man eine Datei komplett durchforsten will, sollte man das Einlesen mit möglichst wenig READ-Systemaufrufen bewerkstelligen, denn so bekommt das Betriebssystem Gelegenheit, den Diskettenzugriff zu optimieren (mehrere Sektoren, gegebenenfalls eine ganze Spur auf einmal lesen), was der Geschwindigkeit zugute kommt.

Sämtliche File-Operationen verlangen nach Angabe der sogenannten Handle-Nummer, die man als Ergebnis der OPEN-Funktion (in D0) erhält. CLOSE schließt eine Datei, entzieht dem Programm die Zugriffsmöglichkeit auf das File, und die Handle-Nummer verliert ihre Gültigkeit.

Nützliche Unterprogramme

Zur Verwendung in anderen Programmen dürften die Unterprogramme zur Zeichen-, String- und Zahlenausgabe interess ınt sein, insbesondere die Routine "decprint". Sie gibt den Inhalt von D0 dezimal aus, entweder mit fester Stellenzahl und führenden Blanks (für Tabellen und ähnliches) oder so kurz wie möglich. Die Routine läßt sich leicht so modifizieren, daß sie nicht auf den Bildschirm, sondern in einen Speicherbereich schreibt.

Es würde mich freuen, wenn der eine oder andere durch diesen Beitrag angeregt würde, selbst Assemblerprogramme auf seinem Atari ST zu entwickeln. Es ist wirklich nicht schwierig, und das Betriebssystem steht einem mit zahlreichen, nützlichen Dienstleistungen zur Seite.

Literatur:
Brückmann, Englisch, Gerits
Atari ST intern
Ein Data-Becker-Buch
ISBN 3-89011-119-X

Michael Schramm

Datei-Info

```
* Assemblerlisting für den ATARI ST

* Inhalt von Dateien untersuchen.
* Ausgabe von Größe, Wort- und Zeilenzahl.

*           Autor: Michael Schramm
*           Computer Kontakt 1986

        .text

_main     movea.l 4(a7),a5         * A5 auf Basepage.
          lea stackend+2,a7        * Stackp. ans Stackende.
          movea.l 32(a5),a6        * A6 auf Parametertext.
          move.l 12(a5),d0         * Länge des Textsegments
          add.l 20(a5),d0          * plus Data-Segmentlänge
          add.l 28(a5),d0          * plus BSS-Segmentlänge
          add.l #256,d0            * plus Basepage-Länge
          move.l d0,-(sp)          * auf den Stack, ebenso
          move.l a5,-(sp)          * die Adr. der Basepage.
          clr.w -(sp)              * Muß sein, Bedeutung?
          move.w #$4a,-(sp)        * Funktionsnummer SETBLCK.
          clr.l d7                 * Alle Flags rücksetzen.
          bsr newline              * Cursor auf Zeilenanfang.
          trap #1                  * Def. Sp.platz reservier.
          add.l #12,sp             * Stack "aufräumen".
          tst.l d0                 * Bei D0 <> 0 ist ein
          bne error                * Fehler aufgetreten.
          tst.b (a6)+              * Falls die Länge des
          bne param_av             * Parameterstrings null
          lea usagetxt,a0          * beträgt, wird ein
          bsr writeln              * Hinweis zur Benutzung
          bra ende                 * ausgegeben.

error     lea errortxt,a0          * Fehlermeldung
          bsr writeln              * ausgeben.
ende      clr.l d7                 * Alle Flags rücksetzen,
          bsr newline              * damit diese
          lea ende_txt,a0          * Ausgaben nur auf dem
          bsr writeln              * Bildschirm erscheinen.
          move.w #1,-(sp)          * CONIN-Funktionsnummer,
          trap #1                  * auf Tastendruck warten.
          clr.w (sp)               * TERM-Funktionsnummer,
          trap #1                  * Programm beenden.

param_av  pea dtaddr               * 44 Bytes langer Buffer.
          move.w #$1a,-(sp)        * SETDTA-Funktionsnummer.
          trap #1                  * Funktion aufrufen, Stack
          addq.l #6,sp             * wieder aufräumen.
          moveq #-1,d0             * Die Größe des noch ver-
          move.l d0,-(sp)          * fügbaren Speicherplatzes
          move.w #$48,-(sp)        * feststellen. Ergebnis
          trap #1                  * wird in D0 zur Verfügung
          addq.l #6,sp             * gestellt. Auf ein Viel-
          andi.l #-512,d0          * faches von 512 abrunden.
          beq error                * Fehler, falls Ergebn. 0.
          move.l d0,d6             * Buffer-Länge -> D6.
          move.l d0,-(sp)          * Länge und Funktionsnr.
          move.w #$48,-(sp)        * auf Stack, Speicherplatz
          trap #1                  * bereitstellen (nur bei
```

```
         addq.l #6,sp              * -1 als Länge erfolgt
         tst.l d0                  * reine Sp.platz-Abfrage).
         bmi error                 * Bei D0 negativ, Fehler.
         move.l d0,a5              * Sonst enthält D0 Anf.adr
                                   * des Sp.pl., diese -> A5.

* Alle vorbereitenden Maßnahmen sind jetzt getroffen. Nun muß
* die Parameterzeile Wort für Wort abgearbeitet werden.

read_par  move.b (a6)+,d0          * Den nächsten Wortanfang
          beq read_par             * suchen, also Nullbytes
          cmpi.b #32,d0            * und Blanks überlesen.
          beq read_par             * Das Stringende ist durch
          bcs ende                 * CR = $0D gekennzeichnet.
          cmpi.b #'-',d0           * Der Bindestrich leitet
          bne n_option             * eine Option ein.
          move.b (a6)+,d0          * Nächstes Zeichen lesen.
          cmpi.b #'P',d0           * Bei P wie Printer Bit 15
          bne no_prn               * im Flag-Reg. D7 setzen.
          bset #15,d7
no_prn    cmpi.b #'M',d0           * Entsprechend Bit 14
          bne read_par             * bei M wie Modem-Port
          bset #14,d7              * setzen. Andere Char.
          bra read_par             * werden ignoriert.

n_option  lea filename,a0          * Raum für Dateiname.
          movea.l a0,a4            * Adresse auch in A4.
          clr.w -(sp)              * Daten für die
          move.l a0,-(sp)          * SFIRST-Funktion.
nxt_byte  move.b d0,(a0)+          * Den Dateinamen aus der
          move.b (a6)+,d0          * Parameterzeile in
          beq name_end             * filename kopieren.
          cmpi.b #33,d0            * Nullbyte, Blank oder CR
          bcc nxt_byte             * zeigen Namensende an.
name_end  clr.b (a0)               * Stringende kennzeichnen.
          subq.l #1,a6             * Parameterpointer dekrem.
          move.w #$4e,-(sp)        * SFIRST aufrufen, sucht
          trap #1                  * nach passendem File.
          addq.l #8,sp
          tst.w d0                 * D0 = 0, falls die Suche
          beq fl_found             * erfolgreich war.
          movea.l a4,a0            * Sonst den Dateinamen
          bsr write                * mit einer entsprechenden
          lea ntfndtxt,a0          * Meldung drucken.
          bsr writeln              * Dann den nächsten
          bra read_par             * Parameter vorknöpfen.

fl_found  lea dtaddr,a3            * Die 44 Bytes ab dtaddr
          lea 30(a3),a0            * enthalten jetzt wesent-
          moveq #14,d3             * liche Informationen
fl_pr_lp  subq.w #1,d3             * über das gefundene File.
          move.b (a0)+,d0          * Zunächst den Namen
          beq flnm_end             * drucken mit nachfol-
          bsr writechr             * genden Blanks, insgesamt
          bra fl_pr_lp             * 15 Stellen.

flnm_end  bsr pr_blank
          dbf d3,flnm_end

          clr.l d0
          move.b #4,d7             * Folgende Ausg. 4stellig.
          move.w 24(a3),d1         * Das Erstelldatum in D1.
          move.b d1,d0             * Bit 15 bis 9 geben die
          andi.b #31,d0            * Jahreszahl minus 1980
          bsr decprd7              * wieder, Bit 8 bis 5
          lsr.l #5,d1              * die Monatszahl und Bit
          move.b d1,d2             * 4 bis 0 den Tag.
          andi.b #15,d2
          bsr pr_datum             * Das Datum wird
          lsr.l #4,d1              * aufbereitet und
          move.b d1,d2             * ausgegeben.
          addi.b #80,d2            * Auf die Zahl für das
          cmpi.b #100,d2           * Jahr wird 80 aufaddiert
          bcs year_ok              * bzw. 20 davon abgezogen
          subi.b #100,d2           * (für Jahreszahlen >=
year_ok   bsr pr_datum             *  2000).
          move.w 22(a3),d1         * Uhrzeit-Byte in D1.
          move.w d1,d0             * Bit 15 bis 11 geben die
          lsr.w #6,d0              * Stundenzahl wieder, Bit
          lsr.w #5,d0              * 10 bis 5 die Minuten.
          bsr decprd7
          moveq #':',d0
          bsr writechr
          move.w d1,d2
          lsr.w #5,d2
          andi.w #63,d2
          bsr pr_2dig
          bsr pr_blank
          move.l 26(a3),d0         * Länge des Files. Bei 0
          beq nxt_file             * nächstes File vornehmen.

          move.l d0,d5             * (Rest-) Länge in D5.
          bsr decpr7dg             * Länge mit 7 Stellen dr.
          movea.l a5,a0            * Buffer-Adr. -> A0.
          lea filename,a1          * Hier steht noch der
          movea.l a0,a2            * Dateiname aus der
dir_loop  move.b (a1)+,d0          * Eingabezeile.
          move.b d0,(a0)+          * Der eingegebene Datei-
          beq dir_end              * name-Suchausdruck wird
          cmpi.b #':',d0           * in den Buffer übertra-
          beq dir_mark             * gen. Gesucht wird der
          cmpi.b #'\',d0           * letzte Doppelpunkt oder
          bne dir_loop             * Backslash (\) im Name.
dir_mark  movea.l a0,a2            * Die Position+1 wird in
          bra dir_loop             * A2 festgehalten und
                                   * hinter diesen Character
dir_end   lea 30(a3),a1            * der von SFIRST bzw.
dirfl_lp  move.b (a1)+,(a2)+       * SNEXT gelieferte File-
          bne dirfl_lp             * name gehängt, denn der
                                   * ist Directory-lokal,
                                   * also ohne Pfad.
          clr.w -(sp)              * Nur zum Lesen öffnen.
          move.l a5,-(sp)          * Adresse des Filenamens.
          move.w #$3d,-(sp)        * Funktionsnummer OPEN.
          trap #1                  * Sofern öffnen möglich
          addq.l #8,sp             * war, bekommt man in D0
          tst.w d0                 * eine 'Handle-Nummer'
          bpl open_ok              * zurückgeliefert.
          lea operrtxt,a0          * Bei D0 negativ,
          bra errbreak             * Fehlermeldung.

open_ok   move.w d0,d4             * Handle-Nr. -> D4.
          lea chrflags,a4          * Die 256 Bytes ab chrflags
          move.w #252,d0           * nullsetzen. Die Bytes
clr_lp    clr.l 0(a4,d0.w)         * werden <> 0 gesetzt wer-
          subq.w #4,d0             * den, falls ein entspr.
          bpl clr_lp               * Byte im File vorkommt.
          clr.l d3                 * Zählt Anzahl der Wörter.
          clr.l d2                 * Zählt Anzahl der Zeilen.
          bclr #31,d7              * : nicht in einem Wort.
read_lp   movea.l a5,a3            * A3 (= aktuelle Leseadr.
          move.l d5,d0             * im Buffer) auf Buf.anf.
          cmp.l d5,d6              * Das Minimum aus Datei-
          bcc len_ok               * restlänge und Buffer-
          move.l d6,d0             * länge in D0. Dateirest-
len_ok    sub.l d0,d5              * länge um D0 verringern.
          tst.l d0                 * Falls alles gelesen,
          beq analysis             * Datenanalyse & -ausgabe.
          move.l a5,-(sp)          * Bufferadresse.
          move.l d0,-(sp)          * Anz. zu lesender Bytes.
          move.w d4,-(sp)          * Handle-Nr. des Files.
          move.w #$3f,-(sp)        * READ-Funktionsnummer.
          trap #1                  * In D0 wird die Anzahl
          addq.l #4,sp             * der tatsächlich gelese-
          cmp.l (sp)+,d0           * nen Bytes zurückgelie-
          addq.l #4,sp             * fert. Falls diese <>
          beq read_ok              * gewünschter Zahl, ist
          lea rderrtxt,a0          * ein Fehler eingetreten.
errbreak  bsr write                * Fehlermeldung ausgeben,
          bra fl_close             * close und nächstes File.

read_ok   clr.l d1                 * In D1 wird Byte für Byte
byte_lp   move.b (a3)+,d1          * des Files übertragen.
          move.b d4,0(a4,d1.w)     * In chrflags berücksicht.
          cmpi.b #10,d1            * LF, CR und Blank werden
          beq found_lf             * als Worttrennzeichen
          cmpi.b #13,d1            * gewertet. Die LF's
          beq new_word             * werden als Anzahl der
          cmpi.b #32,d1            * Zeilen gezählt.
          beq new_word             * Jedes andere Zeichen
          tst.l d7                 * ist Anfang eines Wortes,
          bmi nxt_char             * sofern nicht Bit 31 in
          bset #31,d7              * D7 gesetzt ist. Bit 31
          addq.l #1,d3             * setzen, sobald ein Wort-
nxt_char  subq.l #1,d0             * anfang erkannt ist.
          bhi byte_lp              * Alle gelesenen Bytes.
          bra read_lp              * Nächsten Block lesen.

found_lf  addq.l #1,d2             * Zeilenzähler erhöhen.
new_word  bclr #31,d7              * Wortende erreicht.
          bra nxt_char             * Nächstes Zeichen.

analysis  move.l d3,d0             * Anzahl der Wörter
          bsr decpr7dg             * ausgeben.
          move.l d2,d0             * Anzahl der Zeilen
          bsr decpr7dg             * ausgeben.
          moveq #-1,d0
srch_min  addq.l #1,d0             * Nach dem minimalen im
          tst.b 0(a4,d0.w)         * File vorkommenden Byte
          beq srch_min             * in chrflags suchen,
          move.l d0,d3             * in D3 kopieren und mit
          move.b #5,d7             * fünf Stellen ausgeben.
```



```
          bsr decprd7
          lea intvltxt,a0          * ' <= C <=' ausgeben.
          bsr write
          move.w #256,d0
srch_max  subq.l #1,d0             * In analoger Weise fest-
          tst.b 0(a4,d0.w)         * stellen, welches das
          beq srch_max             * größte Byte ist.
          subq.b #1,d7             * Dieses mit vier Stellen
          bsr decprd7              * ausgeben.
          cmpi.b #10,d3            * F. kleinstes Byte < LF,
          bcs fl_close             * liegt kein reiner ASCII-
          lea 11(a4),a0            * ASCII-Text vor. CR-
          clr.b 13(a4)             * Byte in chrflags := 0.
          moveq #20,d0             * Nur wenn keine weiteren
ascii_lp  tst.b (a0)+              * Steuerzeichen auftreten,
          dbne d0,ascii_lp         * handelt es sich um einen
          bne fl_close             * reinen ASCII-Text.
          lea asciitxt,a0          * Entsprechende Mitteilung
          bsr write                * drucken.
fl_close  move.w d4,-(sp)          * Handle-Nummer.
          move.w #$3e,-(sp)        * CLOSE-Funktionsnummer.
          trap #1                  * Close gibt die Handle-
          addq.l #4,sp             * Nr. etc. wieder frei.

nxt_file  bsr newline              * Zeile abschließen.
          move.w #$4f,-(sp)        * SNEXT, sucht nächstes
          trap #1                  * File, dessen Name zum
          addq.l #2,sp             * eingegeb. Muster paßt.
          tst.l d0                 * Falls D0 <> 0, keine
          beq fl_found             * weiteren Files
          bra read_par             * vorhanden.

pr_datum  moveq #'.',d0            * Zunächst einen Punkt
          bsr writechr             * ausgeben, dahinter den
pr_2dig   moveq #'0',d0            * Inhalt von D0 mit zwei
pr_tst10  cmpi.b #10,d2            * Stellen, gegebenenfalls
          bcc date_gt9             * mit führender Null.
          bsr writechr
date_gt9  move.b d2,d0
          bra decprint

* Das Unterprogramm decprint gibt den Inhalt von D0 dezimal
* aus. Die Zahl wird mit minimaler Stellenzahl gedruckt,
* sofern Bit 29 von D7 zurückgesetzt ist. Anderenfalls
* erfolgt der Druck mir D7.B Stellen.

decpr7dg  move.b #7,d7             * Ausgabe mit 7 Stellen.
decprd7   bset #29,d7              * Ausg. mit D7.B Stellen.

decprint  bclr #30,d7              * Noch keine Ziffer gedr.
          movem.l d0-d3/a0,-(sp)   * Benötigte Reg. retten.
          move.l d0,d2             * Auszugebende Zahl -> D2.
          clr.l d3                 * Anzahl der gewünschten
          move.b d7,d3             * Stellen in D3. Falls Bit
          btst #29,d7              * 29 in D7 nicht gesetzt,
          bne pr_d3_dg             * 7 Stellen, dann jedoch
          moveq #7,d3              * ohne führende Blanks.
pr_d3_dg  lsl.w #2,d3              * D3 in einen Word-Index
          lea pot10tab-4,a0        * für pot10tab verwandeln.
decpr_nx  move.l 0(a0,d3.w),d1     * Eine 10erpotenz lesen.
          moveq #-1,d0             * In D0 zählen, wie oft
decpr_lp  addq.l #1,d0             * die 10erpotenz von dem
          sub.l d1,d2              * Zahlenrest in D2 abge-
          bcc decpr_lp             * zogen werden kann, ohne
          add.l d1,d2              * daß die Zahl neg. wird.
          tst.b d0                 * Den Fall 'nullmal'
          beq dec_pr_0             * gesondert behandeln.

          bset #30,d7              * : bereits Ziffer gedr.
decpr_dg  addi.b.#'0',d0           * Die in D0 enthaltene
decpr_bt  bsr writechr             * Ziffer ausgeben.
          bra decpr_ok             * Evtl. weitere Ziffern.
dec_pr_0  btst #30,d7              * Die Null drucken, falls
          bne decpr_dg             * schon Ziffern gedruckt
          subq.b #1,d1             * oder falls die Einer-
          beq decpr_dg             * stellen zu dr. sind.
          moveq #32,d0             * Sonst ein Blank dr.,
          btst #29,d7              * falls feste Stellen-
          bne decpr_bt             * zahl vorgegeben.
decpr_ok  subq.w #4,d3             * D3 auf nächstkleinere
          bhi decpr_nx             * 10erpotenz und weiter.
          bclr #29,d7              * Für nächsten Aufruf.
          movem.l (sp)+,d0-d3/a0   * Alte Registerinhalte
          rts                      * wiederherstellen.

pr_blank  moveq #32,d0             * Ein Blank drucken.
          bra writechr

writeln   bsr write                * String ausgeben.
newline   moveq #13,d0             * CR ausgeben.
          bsr writechr
          moveq #10,d0             * LF ausgeben.
writechr  movem.l d0-d4/a0-a4,-(sp) * Alle benöt. Reg. retten
          move.w d0,-(sp)          * Char. als Param. auf St.
          move.l #$30002,-(sp)     * Parameter für Ausgabe
          trap #13                 * auf den Bildschirm.
          tst.w d7                 * Falls Bit 15 von D7
          bpl wr_no_pr             * gesetzt ist, Ausgabe
          move.l #$30000,(sp)      * auch auf Drucker; ent-
          trap #13                 * spr. Param. auf Stack.
wr_no_pr  btst #14,d7              * Bei gesetztem Bit 14
          beq wr_no_md             * Ausgabe auch über
          move.l #$30001,(sp)      * Modem-Port; entspre-
          trap #13                 * chende Par. auf Stack.
wr_no_md  addq.l #6,sp             * Stack aufräumen.
          movem.l (sp)+,d0-d4/a0-a4 * Reg. wiederherstellen.
return    rts

write     move.b (a0)+,d0          * Einen String drucken,
          beq return               * der bei A0 beginnt und
          bsr writechr             * durch ein Nullbyte
          bra write                * begrenzt ist.

          .data                    * Liste für decprint:

pot10tab  .dc.l 1,10,100,1000,10000,100000,1000000,10000000

usagetxt  .dc.b 'usage: info (-p) (-m) '
          .dc.b 'filename1 (filename2 ...)',0
errortxt  .dc.b 'error while reserving memory space',0
ende_txt  .dc.b '      <Return>',0
ntfndtxt  .dc.b ' not found',0
operrtxt  .dc.b '  - cannot open',0
rderrtxt  .dc.b '  - read error',0
intvltxt  .dc.b ' <= C <=',0
asciitxt  .dc.b ' ASCII',0

          .bss
stack     .ds.w 199                * Platz für den Stack.
stackend  .ds.w 1
dtaddr    .ds.b 44                 * File-Attribute-Buffer.
filename  .ds.b 60                 * Platz für Dateiname.
chrflags  .ds.b 256                * Flags, ein Byte für je-
                                   * den mögl. Character.
         .end
```

Kreisel

Dieses Programm wurde für den 800 XL geschrieben. Vor einem Probelauf empfiehlt es sich, das Basic-Programm abzusichern. Nach seinem Start werden Sie aufgefordert, eine formatierte Diskette in Station 1 einzulegen. Diese sollte keine DOS-Dateien oder andere Programme enthalten, da sie überschrieben werden können und damit zerstört würden. Wurde eine Diskette eingelegt und RETURN betätigt, so beginnt das Basic-Programm, je 128 DATAs zu lesen, die in Page 6 zwischengespeichert und von dort auf die Diskette (beginnend mit Sektor 1) geschrieben werden.

Wenn man die so erstellte Diskette nun abhängig vom Basic-Programm mit OPTION bootet, erscheint sofort das Spielfeld mit den beiden Kreiseln.

Zwei Personen, die je einen Joystick benötigen, treten gegeneinander an. Jedem Spieler steht eine unbegrenzte Anzahl von Kreiseln zur Verfügung. Verliert er einen, so beginnt er mit einem neuen Kreisel in seiner Spielfeldecke. Siger ist, wer zuerst 33 Punke erreicht hat. Einen Punkt erhält man, wenn der Gegner einen Kreisel verliert oder man selbst ein Kreuz einsammelt. Die Punktanzeige ist wie folgt zu verstehen:

1 Kreuz = 1 Punkt
1 Kreis = 5 Kreuze
1 Gabel = 3 Kreise + 1 Kreuz
1 Quadrat = 2 Gabeln + 1 Kreuz

Das Spielfeld wechselt in bestimmten Zeitabständen seine Farben von Blau nach Dunkelblau, von Dunkelblau nach Rot und von Rot wieder nach Blau. Bewegt man seinen Kreisel auf ein rotes Feld, so geht er verloren. Die weißen Felder wechseln ihre Farbe nicht und können daher gefahrlos benutzt werden.

Solange man den Kreisel nicht bewegt, rotiert er auf der Stelle. Bewegt man ihn aber, so ist er nicht mehr anzuhalten, es sei denn, er kollidiert von hinten mit dem Gegner. Dieser wird dabei aus der Bahn geworfen und ist für einen Moment steuerungsunfähig. Es empfiehlt sich, dieses Manöver am Spielfeldrand auszuführen, um den Gegner ins Abseits oder auf ein rotes Feld zu drängen.

AMC-Verlag

```
10 REM SAVE"D:PQRII.BAS
20 DIM A$(9)
30 A$="hLSd":REM d INVERS EINGEBEN
40 TRAP 660
100 POKE 770,80
110 POKE 769,1
120 POKE 772,0:POKE 773,6
130 POKE 778,1:POKE 779,0
140 GRAPHICS 18
150 POSITION 2,2
160 ? #6;"BITTE FORMATIERTE"
170 POSITION 2,3
180 ? #6;"DISKETTE IN LAUF-"
190 POSITION 2,4
200 ? #6;"WERK 1 EINLEGEN !"
210 POSITION 4,6
220 ? #6;"-dann return-"
230 POKE 764,255
240 IF PEEK(764)<>12 THEN 240
530 FOR I=0 TO 127
540 READ A
550 POKE 1536+I,A
560 NEXT I
570 D=USR(ADR(A$))
630 POKE 778,PEEK(778)+1
650 GOTO 530
660 J=I:FOR I=J TO 127
670 POKE 1536+I,0:NEXT I
680 D=USR(ADR(A$))
690 GRAPHICS 18
700 POSITION 0,2
710 ? #6;"COMPUTER NUN"
720 ? #6;"AUSSCHALTEN"
730 ? #6;"UND SPIEL MIT"
740 ? #6;"option LADEN!"
750 GOTO 750
5000 DATA 0,26,0,64,30,64,169,200
5010 DATA 141,231,2,133,14,169,76,141
5020 DATA 232,2,133,15,169,31,133,10
5030 DATA 169,64,133,11,24,96,96,76
5040 DATA 60,64,169,15,141,198,65,169
5050 DATA 0,141,194,65,141,195,65,141
5060 DATA 196,65,141,197,65,173,31,208
5070 DATA 201,6,208,249,169,0,141,227
5080 DATA 68,141,100,66,141,101,66,169
5090 DATA 0,141,47,2,169,15,141,87
5100 DATA 66,141,88,66,169,75,141,48
5110 DATA 2,169,71,141,49,2,169,72
5120 DATA 141,244,2,169,170,141,192,2
5130 DATA 169,40,141,193,2,169,52,141
5140 DATA 199,2,169,184,141,194,2,32
5150 DATA 102,66,173,223,68,208,18,173
5160 DATA 10,210,41,7,170,189,178,65
5170 DATA 141,223,68,189,186,65,141,220
5180 DATA 68,169,0,133,77,173,31,208
5190 DATA 201,6,208,3,76,60,64,169
5200 DATA 0,141,148,74,141,188,74,141
5210 DATA 189,74,141,228,74,141,229,74
5220 DATA 141,230,74,141,12,75,141,13
5230 DATA 75,141,14,75,141,15,75,141
5240 DATA 136,75,141,176,75,141,177,75
5250 DATA 141,216,75,141,217,75,141,218
5260 DATA 75,141,0,76,141,1,76,141
5270 DATA 2,76,141,3,76,173,100,66
5280 DATA 141,53,65,201,33,144,11,56
5290 DATA 233,33,141,53,65,169,140,141
5300 DATA 136,75,162,0,173,53,65,201
5310 DATA 16,144,15,56,233,16,141,53
5320 DATA 65,169,139,157,176,75,232,76
5330 DATA 244,64,162,0,173,53,65,201
5340 DATA 5,144,15,56,233,5,141,53
5350 DATA 65,169,138,157,216,75,232,76
5360 DATA 12,65,162,0,173,53,65,240
5370 DATA 13,206,53,65,169,137,157,0
5380 DATA 76,232,76,36,65,0,173,101
5390 DATA 66,141,53,65,201,33,144,11
5400 DATA 56,233,33,141,53,65,169,140
5410 DATA 141,148,74,162,0,173,53,65
```



```
5420 DATA 201,16,144,15,56,233,16,141
5430 DATA 53,65,169,139,157,188,74,232
5440 DATA 76,77,65,162,0,173,53,65
5450 DATA 201,5,144,15,56,233,5,141
5460 DATA 53,65,169,138,157,228,74,232
5470 DATA 76,101,65,162,0,173,53,65
5480 DATA 240,12,206,53,65,169,137,157
5490 DATA 12,75,232,76,125,65,173,100
5500 DATA 66,201,33,144,8,169,14,141
5510 DATA 192,2,76,34,64,173,101,66
5520 DATA 201,33,144,8,169,14,141,193
5530 DATA 2,76,34,64,32,135,66,76
5540 DATA 122,64,123,123,99,99,147,147
5550 DATA 124,124,85,155,107,131,107,131
5560 DATA 119,119,0,0,0,0,0,0
5570 DATA 206,199,65,173,199,65,240,1
5580 DATA 96,169,4,141,199,65,173,194
5590 DATA 65,240,3,206,194,65,173,194
5600 DATA 65,141,0,210,141,1,210,173
5610 DATA 195,65,240,9,206,195,65,173
5620 DATA 195,65,141,8,210,141,3,210
5630 DATA 173,196,65,240,9,206,196,65
5640 DATA 173,196,65,141,8,210,141,4
5650 DATA 210,41,15,24,105,96,141,5
5660 DATA 210,173,197,65,240,6,206,197
5670 DATA 65,206,197,65,10,141,200,2
5680 DATA 24,105,60,141,6,210,24,105
5690 DATA 100,141,7,210,173,198,65,240
5700 DATA 14,169,0,141,198,65,170,157
5710 DATA 0,210,232,224,9,208,248,169
5720 DATA 0,141,200,2,173,194,65,201
5730 DATA 13,144,9,173,199,65,105,64
5740 DATA 74,141,200,2,96,0,0,0
5750 DATA 0,0,0,0,0,0,52,194
5760 DATA 100,148,76,100,0,0,173,94
5770 DATA 66,141,221,68,173,97,66,141
5780 DATA 218,68,173,95,66,141,222,68
5790 DATA 173,98,66,141,219,68,169,15
5800 DATA 141,85,66,141,86,66,96,162
5810 DATA 0,160,1,189,91,66,208,15
5820 DATA 189,120,2,201,15,240,8,157
5830 DATA 85,66,169,4,157,91,66,189
5840 DATA 91,66,240,3,222,91,66,189
5850 DATA 85,66,41,1,208,66,222,218
5860 DATA 68,222,218,68,185,218,68,221
5870 DATA 218,68,176,52,189,218,68,56
5880 DATA 233,5,217,218,68,176,41,189
5890 DATA 221,68,56,233,7,217,221,68
5900 DATA 176,30,189,221,68,24,105,7
5910 DATA 217,221,68,144,19,189,85,66
5920 DATA 153,87,66,169,16,153,91,66
5930 DATA 169,15,157,85,66,141,194,65
5940 DATA 189,85,66,41,2,208,66,254
5950 DATA 218,68,254,218,68,185,218,68
5960 DATA 221,218,68,144,52,189,218,68
5970 DATA 24,105,5,217,218,68,144,41
5980 DATA 189,221,68,56,233,7,217,221
5990 DATA 68,176,30,189,221,68,24,105
6000 DATA 7,217,221,68,144,19,189,85
6010 DATA 66,153,87,66,169,16,153,91
6020 DATA 66,169,15,157,85,66,141,194
6030 DATA 65,189,85,66,41,8,208,66
6040 DATA 254,221,68,254,221,68,185,221
6050 DATA 68,221,221,68,144,52,189,221
6060 DATA 68,24,105,7,217,221,68,144
6070 DATA 41,189,218,68,56,233,7,217
6080 DATA 218,68,176,30,189,218,68,24
6090 DATA 105,7,217,218,68,144,19,189
6100 DATA 85,66,153,87,66,169,16,153
6110 DATA 91,66,169,15,157,85,66,141
6120 DATA 194,65,189,85,66,41,4,208
6130 DATA 66,222,221,68,222,221,68,185
6140 DATA 221,68,221,221,68,176,52,189
6150 DATA 221,68,56,233,7,217,221,68
6160 DATA 176,41,189,218,68,56,233,7
6170 DATA 217,218,68,176,30,189,218,68
6180 DATA 24,105,7,217,218,68,144,19
6190 DATA 189,85,66,153,87,66,169,16
6200 DATA 153,91,66,169,15,157,85,66
6210 DATA 141,194,65,138,208,6,168,162
6220 DATA 1,76,139,66,173,87,66,201
6230 DATA 15,240,8,141,85,66,169,15
6240 DATA 141,87,66,173,88,66,201,15
6250 DATA 240,8,141,86,66,169,15,141
6260 DATA 88,66,32,231,68,173,8,208
6270 DATA 41,4,240,16,238,100,66,169
6280 DATA 0,141,223,68,141,2,208,169
6290 DATA 14,141,197,65,173,9,208,41
6300 DATA 4,240,16,238,101,66,169,0
6310 DATA 141,223,68,141,2,208,169,14
6320 DATA 141,197,65,162,0,160,1,189
6330 DATA 0,208,201,0,240,48,189,89
6340 DATA 66,240,8,169,0,157,89,66
6350 DATA 76,155,68,173,226,68,201,1
6360 DATA 240,89,201,3,240,85,201,0
6370 DATA 208,10,189,0,208,41,4,208
6380 DATA 13,76,155,68,189,0,208,41
6390 DATA 2,208,3,76,155,68,169,1
6400 DATA 157,89,66,141,30,208,189,192
6410 DATA 2,141,93,66,24,105,16,157
6420 DATA 192,2,169,47,157,195,65,32
6430 DATA 231,68,222,192,2,189,192,2
6440 DATA 205,93,66,208,242,189,94,66
6450 DATA 157,221,68,189,97,66,157,218
6460 DATA 68,169,15,157,85,66,152,170
6470 DATA 254,100,66,192,0,240,7,162
6480 DATA 1,160,0,76,39,68,169,0
6490 DATA 141,30,208,96,169,0,141,225
6500 DATA 68,141,227,68,162,100,142,218
6510 DATA 68,169,100,141,221,68,142,219
6520 DATA 68,169,120,141,222,68,142,220
6530 DATA 68,169,140,141,223,68,232,232
6540 DATA 142,224,68,32,231,68,96,0
6550 DATA 0,0,0,0,0,0,0,0
6560 DATA 0,0,0,0,0,0,0,173
6570 DATA 227,68,208,84,160,141,162,69
6580 DATA 169,7,32,92,228,169,1,141
6590 DATA 227,68,169,144,141,7,212,169
6600 DATA 3,141,29,208,169,62,141,47
6610 DATA 2,169,1,141,111,2,169,58
6620 DATA 141,192,2,169,186,141,193,2
6630 DATA 169,42,141,194,2,169,6,141
6640 DATA 196,2,169,180,141,197,2,169
6650 DATA 52,141,199,2,169,2,141,226
6660 DATA 68,162,0,138,157,0,147,157
6670 DATA 0,148,157,0,149,232,208,244
6680 DATA 169,0,141,228,68,173,228,68
6690 DATA 240,251,173,226,68,208,10,169
6700 DATA 132,141,197,2,169,48,141,198
6710 DATA 2,173,226,68,201,1,208,10
6720 DATA 169,130,141,197,2,169,132,141
6730 DATA 198,2,173,226,68,201,2,208
6740 DATA 10,169,48,141,197,2,169,132
6750 DATA 141,198,2,173,226,68,201,3
```

```
6760 DATA 208,10,169,132,141,197,2,169
6770 DATA 130,141,198,2,96,238,225,68
6780 DATA 173,225,68,201,150,208,20,238
6790 DATA 226,68,169,0,141,225,68,173
6800 DATA 226,68,201,4,208,5,169,0
6810 DATA 141,226,68,162,0,189,221,68
6820 DATA 157,0,208,24,105,4,157,4
6830 DATA 208,232,224,3,208,239,172,215
6840 DATA 68,162,0,169,0,153,0,148
6850 DATA 200,232,224,16,208,247,185,255
6860 DATA 146,41,8,153,255,146,162,0
6870 DATA 172,218,68,189,139,70,153,0
6880 DATA 148,200,232,224,16,208,244,185
6890 DATA 255,146,9,2,153,255,146,162
6900 DATA 0,172,216,68,169,0,153,0
6910 DATA 149,200,232,224,16,208,247,185
6920 DATA 255,146,41,2,153,255,146,162
6930 DATA 0,172,219,68,189,155,70,153
6940 DATA 0,149,200,232,224,16,208,244
6950 DATA 185,255,146,9,8,153,255,146
6960 DATA 162,0,172,217,68,169,0,153
6970 DATA 0,150,200,232,224,16,208,247
6980 DATA 162,0,172,220,68,189,171,70
6990 DATA 153,0,150,200,232,224,16,208
7000 DATA 244,162,0,189,218,68,157,215
7010 DATA 68,232,224,3,208,245,162,0
7020 DATA 172,230,68,185,187,70,157,139
7030 DATA 70,157,139,70,185,235,70,157
7040 DATA 155,70,185,27,71,157,171,70
7050 DATA 200,232,224,16,208,229,173,230
7060 DATA 68,24,105,16,141,230,68,201
7070 DATA 48,208,10,169,0,141,230,68
7080 DATA 169,1,141,228,68,32,200,65
7090 DATA 76,98,228,0,0,0,0,0
7100 DATA 0,0,0,0,0,0,0,0
7110 DATA 0,0,0,0,0,0,0,0
7120 DATA 0,0,0,0,0,0,0,0
7130 DATA 0,0,0,0,0,0,0,0
7140 DATA 0,0,0,0,0,0,0,0
7150 DATA 0,0,0,4,52,20,22,16
7160 DATA 34,54,54,54,20,20,20,20
7170 DATA 20,0,0,8,8,127,8,9
7180 DATA 3,26,26,26,24,24,24,24
7190 DATA 24,8,8,16,22,0,52,68
7200 DATA 96,44,44,44,12,12,12,12
7210 DATA 12,8,8,4,52,20,22,16
7220 DATA 34,54,54,54,20,20,20,20
7230 DATA 20,0,0,8,8,127,8,9
7240 DATA 3,26,26,26,24,24,24,24
7250 DATA 24,8,8,16,22,0,52,68
7260 DATA 96,44,44,44,12,12,12,12
7270 DATA 12,8,8,0,0,0,0,9
7280 DATA 0,0,8,24,0,0,136,0
7290 DATA 0,0,0,0,0,0,0,0
7300 DATA 10,0,0,0,0,72,0,0
7310 DATA 0,0,0,0,0,0,0,0
7320 DATA 0,12,0,0,40,0,0,0
7330 DATA 0,0,0,112,112,112,68,32
7340 DATA 74,4,4,4,4,4,4,4
7350 DATA 4,4,4,4,4,4,4,4
7360 DATA 4,65,75,71,4,4,4,4
7370 DATA 65,75,71,0,0,0,0,0
7380 DATA 0,0,0,0,0,0,0,0
7390 DATA 0,0,0,0,0,0,0,0
7400 DATA 0,0,0,0,0,0,0,0
7410 DATA 0,0,0,0,0,0,0,0
7420 DATA 0,0,0,0,0,0,0,0
7430 DATA 0,0,0,0,0,0,0,0
7440 DATA 0,0,0,0,0,0,0,0
7450 DATA 0,0,0,0,0,0,0,0
7460 DATA 0,0,0,0,0,0,0,0
7470 DATA 0,0,0,0,0,0,0,0
7480 DATA 0,0,0,0,0,0,0,0
7490 DATA 0,0,0,0,0,0,0,0
7500 DATA 0,0,0,0,0,0,0,0
7510 DATA 0,0,0,0,0,0,0,0
7520 DATA 0,0,0,0,0,0,0,0
7530 DATA 0,0,0,0,0,0,0,0
7540 DATA 0,0,0,0,0,0,0,0
7550 DATA 0,0,0,0,0,0,0,0
7560 DATA 0,0,0,0,0,0,0,0
7570 DATA 0,0,0,0,0,0,0,0
7580 DATA 0,0,0,0,0,0,0,0
7590 DATA 0,0,0,0,1,7,31,85
7600 DATA 1,7,31,127,255,255,255,85
7610 DATA 255,255,255,253,247,223,127,85
7620 DATA 247,223,127,255,255,255,255,85
7630 DATA 255,255,255,253,244,208,64,0
7640 DATA 244,208,64,0,0,0,0,0
7650 DATA 195,195,60,60,60,60,195,195
7660 DATA 20,20,65,65,65,65,20,20
7670 DATA 20,20,20,20,65,65,65,65
7680 DATA 253,253,193,193,193,193,213,213
7690 DATA 63,192,69,68,68,69,192,63
7700 DATA 192,48,17,16,16,16,48,192
7710 DATA 69,204,68,69,64,64,192,69
7720 DATA 69,204,68,69,68,68,204,69
7730 DATA 65,3,1,65,65,65,192,64
7740 DATA 0,0,81,17,17,80,0,3
7750 DATA 0,0,16,16,80,16,48,240
7760 DATA 84,252,68,68,84,68,204,68
7770 DATA 64,243,85,68,64,64,192,64
7780 DATA 69,207,68,68,68,68,207,69
7790 DATA 64,192,0,0,0,0,192,64
7800 DATA 255,255,243,243,255,255,240,240
7810 DATA 15,15,12,12,12,12,15,15
7820 DATA 240,240,48,48,112,112,208,208
7830 DATA 255,255,243,243,255,252,243,243
7840 DATA 252,252,48,48,48,48,252,252
7850 DATA 0,0,0,0,0,0,0,0
7860 DATA 0,0,0,0,0,0,0,0
7870 DATA 0,0,0,0,0,0,0,0
7880 DATA 0,0,0,0,0,0,0,0
7890 DATA 0,0,0,0,1,5,21,85
7900 DATA 1,5,21,85,85,85,85,85
7910 DATA 85,85,85,85,85,84,80,64
7920 DATA 85,84,80,64,0,0,0,0
7930 DATA 170,170,170,170,169,165,149,85
7940 DATA 169,165,149,85,85,85,85,85
7950 DATA 85,85,85,85,85,86,90,106
7960 DATA 85,86,90,106,170,170,170,170
7970 DATA 85,170,170,170,170,170,170,170
7980 DATA 255,255,255,255,255,255,255,85
7990 DATA 170,170,170,170,170,170,170,85
8000 DATA 255,255,255,255,255,255,255,255
8010 DATA 170,170,170,170,170,170,170,170
8020 DATA 255,255,255,255,253,244,208,64
8030 DATA 253,244,208,64,0,0,0,0
8040 DATA 0,0,0,0,1,7,31,127
8050 DATA 1,7,31,127,255,255,255,255
8060 DATA 85,255,255,255,253,245,213,85
8070 DATA 85,245,213,85,85,85,85,85
8080 DATA 85,85,85,85,85,87,95,85
8090 DATA 85,87,95,127,255,255,255,85
```



```
8100 DATA 170,170,170,170,169,167,159,127
8110 DATA 169,167,159,127,255,255,255,255
8120 DATA 255,255,255,255,253,246,218,106
8130 DATA 253,246,218,106,170,170,170,170
8140 DATA 85,170,170,170,169,167,159,127
8150 DATA 85,167,159,127,255,255,255,255
8160 DATA 255,255,255,255,253,246,218,85
8170 DATA 253,246,218,106,170,170,170,85
8180 DATA 85,255,255,255,255,255,255,255
8190 DATA 85,85,85,85,85,85,85,85
8200 DATA 255,255,255,255,253,245,213,85
8210 DATA 253,245,213,85,85,85,85,85
8220 DATA 85,85,85,85,85,87,95,127
8230 DATA 85,87,95,127,255,255,255,255
8240 DATA 0,0,0,0,0,0,0,0
8250 DATA 0,0,0,0,0,0,0,0
8260 DATA 0,0,0,0,0,0,0,0
8270 DATA 0,0,0,0,0,0,0,0
8280 DATA 0,0,0,0,0,0,0,0
8290 DATA 0,0,0,0,0,0,0,0
8300 DATA 0,0,0,0,0,0,0,0
8310 DATA 0,0,0,0,0,0,0,0
8320 DATA 0,0,0,0,0,0,0,0
8330 DATA 0,0,0,0,0,0,0,0
8340 DATA 0,0,0,0,0,0,0,0
8350 DATA 0,0,0,0,0,0,0,0
8360 DATA 0,152,153,154,155,128,156,156
8370 DATA 0,0,0,0,0,0,0,0
8380 DATA 0,0,0,0,0,0,0,0
8390 DATA 0,0,0,0,0,0,0,0
8400 DATA 0,0,0,0,0,0,0,0
8410 DATA 0,0,0,0,0,0,0,0
8420 DATA 0,0,0,0,0,0,0,0
8430 DATA 0,0,0,0,0,0,0,0
8440 DATA 0,0,0,0,0,0,0,0
8450 DATA 0,0,0,0,0,0,141,142
8460 DATA 143,144,144,145,146,147,148,149
8470 DATA 150,151,0,0,0,0,0,0
8480 DATA 0,0,0,0,0,0,0,0
8490 DATA 0,0,0,0,0,0,0,0
8500 DATA 0,0,0,0,0,0,0,0
8510 DATA 0,0,0,0,0,0,0,0
8520 DATA 0,0,0,0,0,0,0,0
8530 DATA 0,0,0,0,0,0,0,0
8540 DATA 0,0,0,0,0,0,0,0
8550 DATA 0,0,0,0,0,0,0,0
8560 DATA 0,0,0,0,0,0,0,0
8570 DATA 0,0,0,0,0,0,0,0
8580 DATA 0,0,0,0,0,0,0,0
8590 DATA 0,0,0,0,0,0,0,0
8600 DATA 0,0,0,0,0,0,0,0
8610 DATA 0,0,48,49,62,62,62,62
8620 DATA 62,62,62,62,62,62,62,62
8630 DATA 62,62,62,62,50,51,35,36
8640 DATA 0,0,0,0,0,0,0,0
8650 DATA 0,0,0,0,0,0,0,0
8660 DATA 48,49,44,44,44,44,44,44
8670 DATA 44,44,44,44,44,44,44,44
8680 DATA 44,44,64,65,35,36,0,0
8690 DATA 0,0,0,0,0,0,0,0
8700 DATA 0,0,0,0,0,0,48,49
8710 DATA 44,44,56,57,41,41,41,41
8720 DATA 41,41,41,41,41,41,58,59
8730 DATA 64,65,35,36,0,0,0,0
8740 DATA 0,0,0,0,0,0,0,0
8750 DATA 0,0,0,0,48,49,44,44
8760 DATA 56,57,45,45,45,45,45,45
8770 DATA 45,45,45,45,54,55,64,65
8780 DATA 35,36,0,0,0,0,0,0
8790 DATA 0,0,0,0,0,0,0,0
8800 DATA 0,0,48,49,44,44,56,57
8810 DATA 45,45,37,38,63,63,63,63
8820 DATA 63,63,63,63,63,63,35,36
8830 DATA 0,0,0,0,0,0,0,0
8840 DATA 0,0,0,0,0,0,0,0
8850 DATA 33,34,63,63,63,63,63,63
8860 DATA 63,63,63,63,39,40,45,45
8870 DATA 54,55,44,44,46,47,0,0
8880 DATA 0,0,0,0,0,0,0,0
8890 DATA 0,0,0,0,0,0,33,34
8900 DATA 66,67,56,57,45,45,45,45
8910 DATA 45,45,45,45,45,45,54,55
8920 DATA 44,44,46,47,0,0,0,0
8930 DATA 0,0,0,0,0,0,0,0
8940 DATA 0,0,0,0,33,34,66,67
8950 DATA 60,61,43,43,43,43,43,43
8960 DATA 43,43,43,43,54,55,44,44
8970 DATA 46,47,0,0,0,0,0,0
8980 DATA 0,0,0,0,0,0,0,0
8990 DATA 0,0,33,34,66,67,44,44
9000 DATA 44,44,44,44,44,44,44,44
9010 DATA 44,44,44,44,44,44,46,47
9020 DATA 0,0,0,0,0,0,0,0
9030 DATA 0,0,0,0,0,0,0,0
9040 DATA 33,34,52,53,42,42,42,42
9050 DATA 42,42,42,42,42,42,42,42
9060 DATA 42,42,42,42,46,47,0,0
```

Für Cassettenbetrieb sind die Zeilen 10-750 durch die folgenden Zeilen zu ersetzen:

```
10 REM SAVE"D:PQRCAS.BAS
20 DATA 104,162,16,76,86,228
30 FOR I=1536 TO 1541:READ A
35 POKE I,A:NEXT I
40 GRAPHICS 18
50 POSITION 0,2
60 ? #6;"    BITTE WARTEN    "
70 FOR I=0 TO 3255:READ A
80 POKE 32768+I,A:NEXT I
90 POSITION 0,2
100 ? #6;"WENN DAS TAPE BEREIT";
110 ? #6;"IST, TASTE  DRUECKEN"
115 POKE 764,255
120 OPEN #1,8,128,"C:"
130 POKE 850,11
140 POKE 852,0:POKE 853,128
150 POKE 856,128:POKE 857,0
160 D=USR(1536)
165 IF PEEK(853)=128+26 THEN 700
170 IF PEEK(852)=0 THEN 200
180 POKE 852,0:POKE 853,PEEK(853)+1
190 GOTO 160
200 POKE 852,128
210 GOTO 160
700 GRAPHICS 18
710 ? #6;"COMPUTER NUN"
720 ? #6;"AUSSCHALTEN"
730 ? #6;"UND SPIEL MIT"
740 ? #6;"option & start"
750 ? #6;"LADEN!"
760 CLOSE #1
770 GOTO 770
```

Puzzler ST

Da leider bisher recht wenig Programme in GfA-Basic bei uns eingetroffen sind, habe ich den "Puzzler" auf den ST umgeschrieben. Er bietet sich besonders für eigene Ein- und Umbauten an. In der abgedruckten Version läuft das Programm nur mit S/W-Monitor. Die Besitzer eines Farbmonitors muß ich leider (vorläufig) enttäuschen.

Zunächst zu den Spielregeln. Mit "Puzzler ST" können sehr einfach Bilder in Schiebe-Puzzles verwandelt werden. Will man nicht auf das eingebaute Testbild zurückgreifen, das aus 60 Zahlenplättchen besteht, so kann man Bilder im "Degas"- oder "Dr. Doodle"-Format laden.

Sobald sich das Bild auf dem Schirm befindet, kann man mit der Option "Programm starten" den Mischvorgang einleiten, vor dem das Programm allerdings nochmals Rückfrage hält. Der Computer zerlegt das Bild dabei in 6×10 bzw. 12×20 (Profi) Quadrate und beginnt, diese über den Bildschirm zu verschieben. Wenn man die Maustaste für einige Sekunden gedrückt hält, kann man dem Einhalt gebieten.

Selbstverständlich ist es jetzt Sache des Spielers, das Bild wieder zu ordnen. Mit der rechten Maustaste kann die Menüleiste aus- und später wieder eingeknipst werden; nur dann lassen sich die Steine auch bewegen. Das Teil, das sich auf das Loch zubewegen soll, wird nur mit der Maus angeklickt, und schon setzt es sich in Bewegung. Um das Programm besonders komfortabel zu gestalten, können auch mehrere Steine nacheinander durch einen einzelnen Knopfdruck bewegt werden. Bei diagonalen Zügen wird zuerst horizontal, dann vertikal verschoben.

Wenn es Ihnen gelingt, das Bild zu restaurieren, so wird Ihre Punktzahl ausgeworfen, die sich wie folgt errechnet: Von der Anzahl der beim Mischen ausgeführten Bewegungen wird die Anzahl der von Ihnen benötigten Bewegungen subtrahiert und mit dem Schwierigkeitsgrad (640/Teilgröße) multipliziert. Als "Profi" erhält man also die doppelte Punktzahl. Negative Werte bedeuten, daß sie mehr Verschiebeoperationen benötigt haben als der Computer. Ist die Punktzahl = 0, so haben Sie die (meist wenigen) Computerzüge einfach umgekehrt.

Wer mit anderen Zeichenprogrammen arbeitet, kann "Puzzler ST" leicht an das jeweilige Bildformat anpassen. Es ist bei den Menüdaten und der Abfrage nur der Name für das entsprechende Format einzusetzen. Das Extenderkürzel (Ext$) sollte dann ebenfalls angepaßt werden. Etwas mehr Aufmerksamkeit erfordert die Variable Formoff, die die Anzahl der Bytes (+1) enthält, die den Bilddaten vorangestellt sind (hier sind normalerweise die Farbinformationen gespeichert). Da zu einem kompletten Bild "nur" 32000 Byte gehören, sind die darüber hinausgehenden Bytes solche Infos. Kürzere Bilderdateien können durch Komprimierung entstehen. Solche Bilder können allerdings nicht direkt geladen werden!

Bei "Puzzler ST" werden einige Leistungsmerkmale des GfA-Basic deutlich. Neben der einfachen und sicheren GEM-Programmierung lassen sich grafikintensive Spiele relativ einfach programmieren. Soweit es sinnvoll ist, wurden alle Funktionen als Prozeduren definiert. Auf diese Weise kommt das ganze Programm ohne ein einziges GOTO aus! Auch von IF/THEN/ELSE/ENDIF wurde reichlich Gebrauch gemacht. "Puzzler ST" enthält keinerlei Maschinensprache- oder C-Unterprogramme. Einige der verwendeten Basic-Tricks möchte ich jedoch nachfolgend erläutern.

Die Teile, die man verschieben will, werden mit der Funktion GET in einen String kopiert. Ein Teil kann dann pixelweise versetzt wieder dargestellt werden. Der überstehende Rest von der vorigen Position wird durch eine schwarze Linie entfernt; so entsteht das neue Loch.

Ein Bild wird zunächst in einen Hilfsstring (P$) geladen. Der 32000 Byte lange Teil ab Formoff wird dann einem String angehängt, der bereits einen 6 Byte langen Header enthält, nämlich die X- und Y-Ausdehnung sowie die Farbe des Rechtecks (jeweils mit MKI$ erzeugt). Mit Put 0,0 kann der so erzeugte String dann auf dem Bildschirm ausgegeben werden.

Der Vergleich zwischen Originalbild und dem Bildschirminhalt erfolgt keineswegs direkt. Zu Beginn werden zwei identische Strings erzeugt (O$ und A$). Bei jeder Verschiebeoperation wird A$ dann sinngemäß mitverschoben. Der Vergleich von O$ und A$ gibt nun Auskunft darüber, ob das Bild fertiggestellt wurde. Dieser Vorgang, der in der Move-Prozedur enthalten ist, arbeitet auch mit den einzigen POKEs in diesem Programm.

Beachten Sie bitte vor dem Abtippen des Programms, daß Ihre Basic-Version aktuell ist. Die neue Ausführung enthält im BLOCK-Untermenü die Funktion "H>ide". Bei der allerersten Fassung funktionierte die Funktion VARPTR nicht richtig! Ansonsten erhalten Sie die aktuelle Version gegen Ihre Originaldiskette direkt bei GfA.

Thomas Tausend


```
' ********************
*********************
*********************
*************
' *P u z z l e r  ST
  -  ein Spielprogra
mm von Thomas Tausend
   (c) TT86 *
' ********************
*********************
*********************
*************
' * Version 1.0 vom 27
.9.86 -  mit besondere
m Dank an Sylvia fr Ih
re Geduld! *
' ********************
*********************
*********************
*************
' * ...ein Megabrain-P
rodukt des Allgu-Teams
 *  Remember the Icon
Grabber!  *
' ********************
*********************
*********************
*************
'
Menu Kill
Ext$="PI3"
For N=1 To 20*12
  O$=O$+Chr$(N)
Next N
A$=O$
Akt=Varptr(A$)
Poke Akt,0
Size=64
Degflag=1
Let Formoff=35
Dim M$(30)
For N=0 To 30
  Read M$(N)
  Exit If M$(N)="ENDE"
Next N
Gosub Menu_on
Deffill 1,2,4
Pbox 0,19,639,399
'
' Hauptschleife
'
Do
  On Menu
  If Mousek=2
    If Gotflag=1
      Gosub Switch_men
u
    Else
      Gosub Wbalarm
    Endif
  Endif
  If Flag=1 And Mousek
=1 And Mousey<384 And
Mixlevel<>0
    X=Mousex
    Y=Mousey
    Ppx=Int(X/Size)
    Ppy=Int(Y/Size)
    Restlevel=Restleve
l+Abs(Ppx-Lpx)+Abs(Ppy
-Lpy)
    Gosub Move
    If O$=A$
      Score=(Mixlevel-
Restlevel)*640/Size
      Put 0,0,Loch$
      B$="GESCHAFFT!|I
hre Punkte: |>>> "+Str
$(Score)+" <<<|"
      If Score>0
        B$=B$+"- prima
!!!"
      Else
        If Score=0
          B$=B$+"(kaum
 gemischt!)"
        Else
          B$=B$+"-schw
ach!"
        Endif
      Endif
      Alert 1,B$,1,"ok
",Donek
    Endif
  Endif
  Exit If Donek=1
Loop
Run
'
' DATAs fuer Menueleis
te
'
Data Info ,  ber Puzzl
er,--------------------
-----,1,2,3,4,5,6,""
Data Bild ,  Degas,  D
oodle,---------------,
  laden,--------------
-
Data  invertieren,---
-------------, Testbil
d,""
Data Spiel ,  starten,
---------------, Prof
i,---------------
Data  beenden,""
Data Hilfe,  Hilfslini
en, Vorlage,"","",END
E
'
' Auswertung der Menle
iste
'
Procedure Mauswert
  Menu Off
  C$=M$(Menu(0))
  If C$="  ber Puzzler
"
    Alert 0,"P U Z Z L
 E R  (c) by TT86|Ein
 Spielprogramm von|Tho
mas Tausend - exklusiv
|fr Computer Kontakt",
1," gelesen ",Dummy
  Endif
  If C$="  beenden"
    End
  Endif
  If C$="  laden"
    Fileselect "\*."+E
xt$,"",Filename$
    If Filename$<>""
      Gosub Load_pic
      Gotflag=1
    Endif
  Endif
  If C$="  starten"
    If Gotflag=1
      Gosub Menu_off
      Put 0,0,Z$
      Gosub Mischen
      Gosub Menu_on
    Else
      Gosub Wbalarm
    Endif
  Endif
  If C$="  invertieren
"
    Gosub Menu_off
    Put 0,0,Z$
    Graphmode 3
    Deffill 1,2,8
    Color 1
    Pbox 0,0,639,399
    Graphmode 1
    Get 0,0,639,399,Z$
    Gosub Menu_on
  Endif
  If C$="  Degas"
    Ext$="PI3"
    Degflag=1
    Ddlflag=0
    Let Formoff=35
    Gosub Menu_off
    Gosub Menu_on
  Endif
  If C$="  Doodle"
    Ext$="DOO"
    Degflag=0
    Ddlflag=1
    Let Formoff=0
    Gosub Menu_off
    Gosub Menu_on
  Endif
  If C$="  Testbild"
    Lpx=0
    Lpy=0
    A$=O$
    Mixlevel=0
    Gotflag=1
    Deffill 1,2,2
    Gosub Menu_off
    Pbox 0,0,639,399
    Gosub Helpdraw
    Deftext 1,17,0,32
    For S=0 To 9
      For Z=0 To 5
        T$=Chr$(Z+48)+
Chr$(S+48)
        Text S*64+12,Z
*64+44,40,T$
      Next Z
    Next S
    Deftext 1,0,0,6
    Deffill 0,0
    Pbox 0,385,639,399
    Color 1
    Text 20,396,600,"
PUZZLER ST - (c) 1986
by Thomas Tausend - mi
t besonderem Dank an S
ylvia!"
    Get 0,0,639,399,Z$
```

```
    B$=Z$
    Gosub Menu_on
    Deftext 1
  Endif
  If C$="  Profi"
    If Size=64
      Size=32
    Else
      Size=64
    Endif
    Gosub Menu_off
    Gosub Menu_on
  Endif
  If C$="  Hilfslinien
"
    If Botflag=1
      Gosub Menu_off
      Put 0,0,Z$
      Gosub Helpdraw
      Get 0,0,639,399,
Z$
      Gosub Menu_on
    Else
      Gosub Wbalarm
    Endif
  Endif
  If C$="  Vorlage"
    If Botflag=1
      Gosub Menu_off
      Get 0,0,639,399,
Z$
      Put 0,0,B$
      While Mousek<>1
      Wend
      Put 0,0,Z$
      Gosub Menu_on
    Else
      Gosub Wbalarm
    Endif
  Endif
Return
'
' Menleiste an/abschal
ten
'
Procedure Switch_menu
 While Mousek<>0
 Wend
 If Flag=0
   Gosub Menu_off
   Put 0,0,Z$
 Else
   Get 0,0,639,399,Z$
   Gosub Menu_on
  Endif
Return
'
' Teil(e) bewegen
'
Procedure Move
  Partx=Ppx*Size
  Party=Ppy*Size
  Lx=Lpx*Size
  Ly=Lpy*Size
  Xdir=Sgn(Ppx-Lpx)
  If Xdir<>0
    For Pp%=Lpx+Xdir T
o Ppx Step Xdir
      Tempx=Pp%*Size
      Get Tempx,Ly,Tem
px+Size-1,Ly+Size-1,Pa
rt$
      For N%=Tempx-Xdi
r To Tempx-Xdir*Size S
tep -Xdir
        Put N%,Ly,Part
$
        If Xdir=-1
          Draw N%-1,Ly
 To N%-1,Ly+Size-1
        Else
          Draw N%+Size
,Ly To N%+Size,Ly+Size
-1
        Endif
      Next N%
      Poke Akt+20*Lpy+
(Pp%-Xdir),Peek(Akt+20
*Lpy+Pp%)
      Poke Akt+20*Lpy+
Pp%,1
    Next Pp%
  Endif
  Lpx=Ppx
  Ydir=Sgn(Ppy-Lpy)
  If Ydir<>0
    For Pp%=Lpy+Ydir T
o Ppy Step Ydir
      Tempy=Pp%*Size
      Get Partx,Tempy,
Partx+Size-1,Tempy+Siz
e-1,Part$
      For N%=Tempy-Ydi
r To Tempy-Ydir*Size S
tep -Ydir
        Put Partx,N%,P
art$
        If Ydir=1
          Draw Partx,N
%+Size To Partx+Size-1
,N%+Size
        Else
          Draw Partx,N
%-1 To Partx+Size-1,N%
-1
        Endif
      Next N%
      Poke Akt+20*(Pp%
-Ydir)+Lpx,Peek(Akt+20
*Pp%+Lpx)
      Poke Akt+20*Pp%+
Lpx,1
    Next Pp%
  Endif
  Lpy=Ppy
Return
Procedure Load_pic
  Lpx=0
  Lpy=0
  A$=D$
  Mixlevel=0
  P$=String$(32200," "
)
  P=Varptr(P$)
  Bload Filename$,P
  B$=Mki$(639)+Mki$(39
9)+Mki$(1)+Mid$(P$,For
moff,32000)
  Z$=B$
  Gosub Menu_off
  Put 0,0,B$
  Gosub Menu_on
Return
Procedure Mischen
  Alert 1,"mischen",1,
" OK | Nein ",Dummy
  If Dummy=1
    If Mixlevel=0
      Get 0,0,Size-1,S
ize-1,Loch$
      Deffill 1,2,8
      Pbox 0,0,Size-1,
Size-1
    Endif
    Do
      Ppx=Random((640/
Size)-1)
      Ppy=Random((384/
Size)-1)
.     Mixlevel=Mixleve
l+Abs(Ppx-Lpx)+Abs(Ppy
-Lpy)
      Gosub Move
      Exit If Mousek<>
0
    Loop
    Get 0,0,639,399,Z$
  Endif
Return
Procedure Menu_on
  Flag=0
  Menu M$()
  On Menu  Gosub Mausw
ert
  Menu 11,Deqflag
  Menu 12,Ddlflag
  Menu 23,Sgn-(Size/32
-2)
Return
Procedure Menu_off
  Menu Kill
  Flag=1
Return
Procedure Helpdraw
  For S=0 To 640 Step
Size
    Draw S,0 To S,399
  Next S
  For Z=0 To 384 Step
Size
    Draw 0,Z To 639,Z
  Next Z
Return
Procedure Wbalarm
  Alert 1,"Bitte laden
 Sie erst ein|Bild ein
 oder verwenden|Sie di
e Menoption|'TESTBILD'
!",1," AHA! ",Dummy
Return
```






Leserecke!

Wer sitzt nicht gelegentlich haareraufend vor dem Computer, und es gelingt einfach nicht, das dritte Bild zu überleben? Oder weil das Anwenderprogramm an entscheidender Stelle genau das nicht tut, was wir erwarten?

Mit Ausdauer oder Glück gelingt aber auch manche Entdeckung, die Anleitung oder Handbuch verschweigen. Damit aber nicht jeder in seinem stillen Kämmerlein das Rad neu erfinden muß, wollen wir in der Leserecke Hilfesuchende und Experten unter unseren Leser zusammenbringen.

Schildern Sie uns die Probleme mit Ihren Programmen; schreiben Sie uns Ihre Entdeckungen. Ihre Fragen und Anregungen werden im ATARI *magazin* veröffentlicht.

Spiele-Ecke

Hier dürfen Sie Hilfe bei Ihren Expeditionen durch den Dschungel der Spiele erwarten

Willkommen, Abenteurer! In dieser Sparte werden wir spielfreudigen Lesern Ratschläge geben. Heute begeben wir uns mit dem Diskfixer auf Drachenjagd.

Diskettenmonitore sind für den Adventurefan unentbehrlich. Auch ein normaler Maschinensprachemonitor, der die Möglichkeit bietet, einzelne Sektoren in den Speicher zu laden und zu editieren, wie z.B. der Monitor XL, kann gute Dienste leisten. Geduldiges "Durchblättern" der in der Regel 720 Sektoren pro Diskseite fördert in vielen Fällen den Wortschatz eines Adventures, das Charaktersheet eines Rollenspiels oder andere nützliche Dinge zutage. Schauen wir uns nun die Playerdisks von Ultima II und III näher an.

Viele Spieler haben Ultima II frustriert beiseite gelegt, weil sie den einen oder anderen wichtigen Gegenstand (z.B. das Blue Tassel) nicht finden konnten oder es leid waren, ständig den von Lord British erdachten Kreaturen zu unterliegen. Werfen wir einen Blick in die Sektoren 16 und 17 (dezimal). Bei einer mit einem Charakter beschriebenen Disk müßten die ersten Bytes den Namen der Spielfigur in inversen Buchstaben wiedergeben (natürlich nur, wenn Ihr Diskfixer die Sektorbytes als Charakter und nicht als Zahlen ausgibt). Die weitere Sektorbelegung ist dem Kasten zu entnehmen.

In Ultima III sind die Sektoren 545 ff interessant. Hier kann ein toter Charakter per Tastendruck ins Leben zurückgerufen werden, ohne daß Zeit und Gold verlorengeht. Durch Erhöhung der Erfahrungspunkte läßt sich die mühselige Anfangsphase des Spiels leicht umgehen. Ebenso ist es möglich, Spielfiguren mit Gegenständen auszustatten, die sie laut Anleitung nicht besitzen dürfen (z.B. Zauberer mit Plate-Armour). Logischerweise darf man eine so erworbene Ausrüstung niemals ablegen, da das Programm ein Wiederaufnehmen verhindern würde.

Das Mapping der Sektoren 8 ff auf der Britannia-Disk von Ultima IV überlasse ich interessierten Lesern. Nur soviel: Ihre Spielfigur nimmt den Platz von dem der 8 Charaktere auf der Disk ein, der denselben Beruf hat. Dadurch ist es z.B. unmöglich, in der Stadt Britain den Barden Iolo anzuwerben, wenn man selbst Barde ist. Übrigens habe ich mit meiner Version von Ultima IV (US GOLD) ein Problem. Mir ist es bis jetzt noch nicht gelungen, trotz unzähliger Treffer ein einfaches Monster zu vertreiben oder zu erlegen. Bei der C-64-Version eines Bekannten besiegt man die Gegner schon als schwacher Magier. Hat jemand ähnliche Erfahrungen gemacht?

Weiter geht's mit unserer Reise durch die Disksektoren. Die sinnvollste Anwendung des Diskfixers ist für Abenteurer die Suche nach dem Wortschatz des Spiels. Gerade bei Grafikadventures wird das Vokabular aus Speicherplatzmangel zugunsten der aufwendigsten Bilder reduziert. Mancher Spieler mag zwar die Lösung eines Problems kennen, ist aber nicht imstande, diese in die Worte zu kleiden, die der Parser versteht. Meist gibt sich dann der User geschlagen, und das Adventure wandert in die Schublade. Deshalb haben wir den Wortschatz von "Sereamis" abgedruckt, gefunden in den Sektoren 475 ff der Programmdisk.

Komplizierter gestaltet sich das Erforschen von "Lucifer's Realm". Dieses Adventure weist zwar einen guten Plot und eine ausgezeichnete Grafik auf, andererseits aber einen schlechten Parser. Das Programm versteht manches Wort nur in einem Raum, im nächsten nicht. Die Sektoren 248 ff und 321 ff bringen Licht in diese Angelegenheit. Neben dem permanenten Wortschatz erscheint der Parser-Wortschatz auf den ersten Blick wie unverständliches Kauderwelsch. Bei genauem Hinsehen kann man aber erkennen, daß z.B. "gvecrwn" eine komprimierte Form von "give crown" ist. Der Parser sucht also nur bestimmte Buchstabenkombinationen in der Eingabe des Spielers. So wird klar, daß er an einem Ort "turn handle" versteht, wenige Züge später aber weder mit "turn" noch mit "handle" etwas anzufangen weiß. Durch den Parser-Wortschatz können viele Rätsel von "Lucifer's Realm" gelöst werden.

Immer wieder wird nach Unsterblichkeits-Pokes für Spiele gefragt. Beim Atari ist eine solche Manipulation durch die eingebaute Auto-Run-Funktion nicht so leicht wie etwa beim C 64 oder Spectrum. Es besteht zwar die Möglichkeit, Programme mit einem Diskfixer vor dem Laden abzuändern, was aber erstens sehr mühsam ist (Programm ändern, laden, es stürzt ab usw.) und zweitens die Gefahr in sich birgt, geschützte Programme irreversibel zu schädigen. Eine andere Methode verspricht mehr Erfolg, kostet aber etwas Geld. Mit einem Monitor als Hardwarezusatz, wie etwa "Bibomon" vom Compy-Shop, läßt sich der Programmablauf nach Belieben anhalten. Nachdem man die richtigen Speicherplätze mit den richtigen Bytes belegt und die Startadresse herausgefunden hat, kann man das Spiel mit vielen Extraleben fortsetzen. Manche Software reagiert zwar allergisch auf solche Computeranhängsel, aber für die meisten Programme müßten sich Superpokes finden lassen.

Soviel für heute. Nächstes Mal geht es um das Rollenspielepos "Alternate Reality" und Tricks zu diversen Martial-Arts-Games.

Frank Emmert

Sereamis-Wortschatz

(Bei den Verben sind auch die Imperative möglich; z.B. sehen/sieh.)

Sehen, betrachten, schauen, gehen, laufen, öffnen, kaufen, binden, befestigen, schlingen, verbeugen, neigen, nehmen, legen, trinken, geben, schenken, füllen, verschieben, schieben, ziehen, stehen klettern, steigen, setzen, schlagen, werfen, leeren, verstecken, verbergen, sprechen, sagen, rufen, töten, ermorden, spielen, füttern, essen, wünschen, erbitten, schließen, zünden, streicheln, lösen entfernen, pflücken, bumsen, ficken, küssen, streuen

Nord(en), Süd(en), Ost(en), West(en), rauf, oben, hinauf, runter, unten, hinunter

Sandale, Stiefel, Skelett, Zechinen, Schals, Tücher, Schlange, Pralinen, Gewand, Pulver, Pistazien, Körbe, Flöte, Dolch, Schlüssel, Suppe, Teller, Lampe, Teppich, Blume, Fleisch, Kohl, Brot, Koran, Buch, Eimer, Seil, Orange, Wein, Stein, Orchidee, Knochen, Panther, Katze, Sultanin, Küche, Dich, Mekka, Wärter, Wächter, Bank, Leilah, Tisch, Falltür, Dogge, Hund, Bestie, Koch, Vorhang, Gardine, Odaliske, Fenster, Tür(e), Truhe, Schrank, Gang, Decke, Wasser, Papagei, Vogel, Kette, Geist, Sarg, Wasserfall, Ecke, Bettler, Händler, Skorpion(e), Schacht, Wand, Mauer(n), Haken, Allah-Jahu-Allah-Yaha

rechten, linken ovale(n), runde(n), eckige(n), fest, los, ein, zu, ab, weg, auf, hin, aus, an

Lucifer's Realm

Permanenter Wortschatz: Look, kill, climb, read, talk, open, unlock, eat, move, dig, drink, taste, jump, push, throw, turn, tell, listen, get, drop, u(p), d(own), n(orth), s(outh), e(ast), w(est)

save game, quit, load game, graphic club, crown, crystal, chain, oilcan, dagger, sword, rope, skull, lamp, disk sunglasses, wax, coins.

Parser-Wortschatz. Jedes Kommando wird nur an bestimmter Stelle im Spiel verstanden.

Getup, getout, looptrn, looldg, pllch, ynkchan, clmdrn, godran, clmbdwn, twsthndl, turnhnd, turnvlve, oprigh, gorigh, entrrgh, opnleft, goleft, entrleft, lormns, exmnrmns, hldnse, clmbopng, goopnng, clmbundrrck, clmbhole, Hitlr, Adolph, Lucifer, Satan, jonary, volnter, joinhtlr, loonazi, lucfge, lobloo, thrskll, brekslI, smshskll, tierpe, hookrpe, ropspik, fastnrpe, bretvpr, inhale, smellvpr, loogurd, loojim, loorvrnd, oilwing, oildoor, rmvglss, takoffglss, clmchain, gochain, upchain, hookch, pullchan, remvechan, putchan, fastnchan, deecl, statu, disk, lohtlr, hrchan, tosschan, hookchan, chanhook, loolght, loobeam, cutcnvs, slitcnvs, tearcnvs ripcnvs, hoochn, pullchan, remvchan, insrtsw, swdslit, stickswrd, sitdwn, sitpar, barslt, bardoor, crystslt, crynoor, cutsck, tearsck, ripsck, lotee, lowll, climbcrk, gocrck, crwlcrk, entrcrk, gocrack, clmbcrk, crwlcrk, entrcrk, givcon, handcoi, givmony, givbrbe, knock, gvecrwn, handcrwn, cofess, pit, watr, objct, bllb, rock, offcr, chst, man, clff, armi, vapr, wll, imges, pllr, spdrs, cnvs, vent, stne, snke, crps, altr, door, crack.

Frank Emmert

Ultima-II-Charaktersheet

(Alle Zahlen dezimal, der Buchstaben-Hinweis bezieht sich auf den ATASCII-Code des Bytes)

Sektor 16

Sektorbyte

- 00-15 Name des Charakters in inversen Buchstaben
- 16 Geschlecht (invers M = männlich, invers F = weiblich)
- 17 Beruf (0 = Fighter, 1 = Cleric, 2 = Wizard, 3 = Thief)
- 18 Rasse (0 = Human, 1 = Elf, 2 = Dwarf, 3 = Hobbit)
- 19, 20 Position auf Landkarte
- 21 Strength
- 22 Stamina
- 23 Wisdom
- 24 Agility
- 25 Charisma
- 26 Intelligence
- 27 Highbyte für HP
- 28 Lowbyte für HP
- 29 Highbyte für Food
- 30 Lowbyte für Food
- 32 Highbyte für EP
- 33 Lowbyte für EP
- 34 Highbyte für Gold
- 35 Lowbyte für Gold
- 43 Weapon (ready) (0 = Hands, 1 = Dagger ...9 = Quicksword)
- 44 Armour (worn) (0 = Skin, 1 = Cloth ...5 = Reflect)
- 45 Spell (ready) (0 = none, 1 = Light...9 = Kill)
- 46 Torches
- 47 Keys
- 48 Tools
- 64 Daggers
- 65 Maces
- 66 Axes
- 67 Bows
- 68 Swords
- 69 Great Swords
- 70 Light Swords
- 71 Phasers
- 72 Quickswords
- 96 Clothes
- 97 Leathers
- 98 Chains
- 99 Plates
- 100 Reflects
- 101 Power Armours

Sektor 17

- 1 Lights
- 2 Down Ladders
- 3 Up Ladders
- 4 Passwalls
- 5 Surfaces
- 6 Prayers
- 7 Magic Missiles
- 8 Blinks
- 9 Kills
- 32 Rings
- 33 Wands
- 34 Staffs
- 35 Boots
- 36 Cloaks
- 37 Helms
- 38 Gems
- 39 Ankhs
- 40 Red Gems
- 41 Skull Keys
- 42 Green Gems
- 43 Brass Buttons
- 44 Blue Tassels
- 45 Strange Coins
- 46 Green Idols
- 47 Try-Lithiums

Ultima-III-Charaktersheet

Die Sektoren 545-554 enthalten alle verfügbaren Spielfiguren, 555 und 556 nur die 4 Charaktere der Party. Jeder Sektor birgt 2 Charaktere.

1. Charakter: Byte 00-63; 2. Charakter: Byte 64-127.

Sektorbyte

- 0-13 Name des Charakters in inversen Buchstaben
- 14 Special:
  - Bit 0 = Card of Love,
  - Bit 1 = Card of Sol,
  - Bit 2 = Card of Moons,
  - Bit 3 = Card of Death,
  - Bit 4 = Mark of Force
  - Bit 5 = Mark of Fire,
  - Bit 6 = Mark of Snake,
  - Bit 7 = Mark of Kings
  - (Ein Wert von 33 in diesem Sektorbyte würde z.B. bedeuten, daß der Charakter die Card of Love und das Mark of Fire besitzt: Bit 0 gesetzt (1) + Bit 5 gesetzt (32) ergibt 33)
- 15 Torches
- 16 Wert 255 bedeutet, daß der Charakter Mitglied der Party ist.
- 17 Zustand des Charakters: invers G = good, invers P = poisoned, invers D = dead
- 18 Strength
- 19 Dextery
- 20 Intelligence
- 21 Wisdom
- 22 Rasse (0 = Human...4 = Fuzzy)
- 23 Beruf (0 = Fighter... 10 = Ranger)
- 24 Geschlecht: invers F = weiblich, invers M = männlich, invers O = other
- 26 Highbyte für HP
- 27 Lowbyte für HP
- 28 Highbyte für HM
- 29 Lowbyte für HM
- 30 Highbyte für Exp. Points
- 31 Lowbyte für Exp. Points
- 35 Highbyte für Gold
- 36 Lowbyte für Gold
- 37 Gems
- 38 Keys
- 39 Powders
- 40 Armour (worn) (0 = Skin... 7 = Exotic)
- 41 Clothes
- 42 Leathers
- 43 Chains
- 44 Plates
- 45 +2 Chains
- 46 +2 Plates
- 47 Exotics
- 48 Weapon (ready) (0 = Hands... 15 = Exotics)
- 49 Daggers
- 50 Maces
- 51 Slings
- 52 Axes
- 53 Bows
- 54 Swords
- 55 2-Hand-Swords
- 56 +2 Axes
- 57 +2 Bows
- 58 +2 Swords
- 59 Gloves
- 60 +4 Axes
- 61 +4 Bows
- 62 +4 Swords
- 63 Exotic Weapons

Kaufe und tausche Programme + Spiele für Atari 800 XL. Liste an M. Götz, Alb.-Schw.-Allee 59, 6200 Wiesbaden

●●● ATARI ●●●
Verkaufe 1025 Matrix-Drucker, guter Zustand, für 299 DM! Suche deutsche Betriebsanleitung für Star NL-10. ☎ 02 34 / 33 01 80

Achtung Atarianer! Suche und biete Software für Atari 800 XL (Disk bevorzugt). Reiner Kuen, Schmiedstr. 8, 8901 Wehringen

800 XL + 1050 + Software + Bücher für 500 DM zu verkaufen. ☎ 02 11 / 70 87 21

Atari XL Software auf Cassette, keine Raubkopien, Programme ab 50 Pf!! Liste gegen 80 Pf Rückporto bei Michael Wagner, Altenburger Str. 64, 6320 Alsfeld

Floppy 810 Happy 450.– DM, 1050er Happy Board 160.– DM, Wiesemann Interface 160.– DM. ☎ 0 46 62 / 54 32

●●● Atari XL / XE ●●●
Verkaufe Orig. Software für Atari auf Cass. Z.B. Mediator, Jump Jet, Colossus Chess, Assylum, Spitfire Ace, Tank Com., Super Zax, Sidewinder, Electra Clide usw., je 20 DM. Liste bei Kai-Eric Weber, Brüderweg 151, 5900 Siegen. Suche Tauschpartner im Raum Siegen

ATARI 1029 Matrixdrucker mit Grafikdruckprogramm (Disk), ca. 11 Monate alt, 1a-Zustand, in Originalverpackung: 345.– DM. ☎ 09 21 / 8 11 55 (nach 18 Uhr).

ATARI 800 XL: Wer sendet mir zu "THE GOONIES" die Spielanleitung? Andreas Krebs, Bahnhofsiedlung 1, 8941 Buxheim

Achtung Atari-Freaks: Verkaufe Drucker SEIKOSHA GP 100 AT für 150.– DM. H. Böhling, Speestr. 4, 4000 Düsseldorf, ☎ 02 11 / 28 31 07

Suche für 800 XL Spiele jeder Art. Besonders Schach! ☎ 06 71 / 3 14 62

●●● ATARI ●●●
Matrix-Drucker ATARI 1029 neuwertig + 2 Hardcopypr. + Etikettendruckpr. zu verkaufen, VHB 350 DM. M. Brucher, ☎ 0 76 43 / 56 76

●●● SUCHE DRINGEND ●●●
Handbuch für MAC/65, Vers. 1.00, Cartridge-Version; kaufen oder leihen, Angebote erbeten. Ferner gesucht: Spielanleitungen (Kopien) von: Countdown, D-Bug, Simulated Comp., Trains. B. Streu, Turnsee 1, 7800 Freiburg, ☎ 07 61 / 7 48 61

Platine für Atari-MIDI zu verkaufen. Platine allein 19.– DM, bestückt 69.– DM. Bei Vorkasse keine Gebühr. P. Böhm, Ronsperger Str. 40, 7070 Schw. Gmünd

Suche Cass.-Software für Atari 800 XL. Angebote an Jürgen Ayasse, Zehnthof-Str. 8, 7530 Pforzheim

●●● SUCHE FÜR ATARI 800 XL ●●●
Amateurfunk-Fernschreib-Programm auf Cassette. Angebot an Michael Klemp, DL7AAZ, Richard-Tauber-Damm 27d, 1000 Berlin 48, ☎ 0 30 / 7 42 39 49

● (800 XL) ●● ATARI ●● (800 XL) ●
Verkaufe, tausche, kaufe Atari-Spiele (Maschinensprache u. Basic, z.B. H.E.R.O., Decathlon u. viele mehr). MS kosten 3-6 DM, BS gibt es ab 0.25 DM. Verkaufe billig MSX-Computer. Stephan Bihler, Ulmer Str. 1, 8943 Babenhausen!!! (nur Cassette)

Verkaufe Atari-1050-Floppy mit DOS-3-Diskette, Design-Master und einer Diskette mit Spielen für nur 350 DM, Datasette XC 11 für 80 DM, 800 XL zum Ausschlachten 100 DM. ☎ 0 63 01 / 40 16 ab 17 Uhr anrufen

Verk. für ATARI 800 XL / 130 XE 3 Disketten mit 50 Programmen, 32 Spiele + 18 × Sound + Grafik + Beschreibung für 80.– DM. 1 Disk mit 15 Anwendungen 30.– DM, bar oder Scheck. Suche Video-Archiv + LST.-Ausgleich-86-Programm. H. Nagel, Josef-Maria-Olbricht-Str. 34, 4000 Düsseldorf 1

Verk. 800 XL, 128 K-RAM + Floppy 1050 + zwei Joysticks + Software (Bruce Lee, Chess, Summergames, Int. Karate, Cyrus u.a.) zusammen VB 600 DM. ☎ 0 54 21 / 18 77. Verk. außerdem Soundmachine, Hexenküche je 15 DM und eine Tischorgel PS-3 von Yamaha: 475 DM (9 Instr. u. 4 Rhythmen)

●●● ATARI 400/800/XL/XE ●●●
Verkaufe Spiele + Anwendungen. Z.B. von ECA, Synapse, Lucas Film auf Disk, Cass. und Modul. Preise von 5 bis 30 DM. Liste anfordern. ☎ 0 23 04 / 4 15 10. Verkaufe außerdem Atari 1029 Drucker mit Hardcopy-Programm (Top-Zustand), VB 380 DM

ATARI XL/XE ●● Suche: Summergames II + Wintergames. ☎ 0 93 72 / 59 43, Bernd verlangen!

●●● Moin Atari-XL/XE-Freunde!! ●●●
NEU: Quiz – Das Spiel für die gesamte Familie. Über 350 Fragen. Diskette 20.– DM (Schein) oder per NN (+ 4.– DM). **NEU:** "Das Kreuz", ein illustriertes Textadventure mit Karte etc. (D: 20.– DM) Info gegen –.80 DM bei GbR K+K, Bikkernstr. 161, 4650 Gelsenkirchen

Suche Drucker für Atari 800 XL und Softlearning Englisch. Angebote an: Günter Streit, Hohenstein 42, ☎ 02 02 / 55 08 14, 5600 Wuppertal 2

● ATARI ● ATARI ● ATARI ● ATARI ●
Kaufe, tausche und verkaufe Spiele für 4 DM. Sofort Liste anfordern bei Michael Mueller, Am Rundacker 2, 6222 Geisenheim. Nur für Atari XL.

Suche Programme für Atari 800 XL auf Cass./Disk. Bitte schreibt an Hermann Bock, Breitenweg 70, 2800 Bremen 1

Suche Floppy 1050 für Atari. Zahle bis 200 DM. ☎ 0 87 43 / 25 16

ATARI 260 ST: 1 Megabyte Aufrüstung, steckbare Platine ohne Lötarbeiten, Epromsteckkarte und User-Port-Karte. **Für 400/800:** OS Rev. B und es laufen alle Programme. **Happy 1050** und Highchip. ☎ 02 02 / 40 28 14

●●● ATARI ST ●●●
Software für Erwachsene (Pornos), Biorhythmus, Reversi, Solitaire uvm. Info anfordern bei: Soft, Pf. 1619, 4830 Gütersloh

●●● Staubkiller ●●●
Abdeckhauben für Atari 600/800 XL / 1050, C 64 / 1541, Klarsicht 14.95 DM. Kunstleder 19.95 DM. VK oder NN. Bei M. Kühn, Ulmenstr. 16, 4100 Duisburg 17, ☎ 0 21 36 / 3 19 77, nach 17 Uhr. Bei Vorkasse keine Gebühr!

Bestellschein für Kleinanzeigen

Bitte veröffentlichen Sie in der nächsten Ausgabe folgende Anzeige:

Meine Anzeige soll in 1 ☐ 2 ☐ 3 ☐ Ausgaben erscheinen.

Vor- und Zuname

Straße

PLZ/Ort

Datum Unterschrift

Bitte ankreuzen:

- ☐ **Private Kleinanzeige: pro Zeile 1.– DM**
- ☐ **Gewerbliche Eigenentwicklungen: pro Zeile 3.– DM**
- ☐ **Gewerbliche Kleinanzeige: pro mm 2.40 DM**

Bei mehreren Kleinanzeigen bitte Bestellschein kopieren. Den Betrag in Briefmarken oder als Scheck zusammen mit der Kleinanzeige einsenden an:

ATARI *magazin*
Redaktion
Postfach 1640, 7518 Bretten

GEM für den Atari 520 ST

Von Josef und Gerhard Steiner
Verlag Markt & Technik
350 Seiten, 52,– DM
ISBN 3-89090-230-8

Was wäre der Atari ST ohne seine Benutzeroberfläche GEM, die seit dem Macintosh als Garant für einen anwenderfreundlichen Computer gilt. Das vorliegende Buch erhebt nun den Anspruch, sich an je-



den zu wenden, der mit GEM auf irgendeine Weise zu tun hat, sei es durch Programmierung oder reine Anwendung. Aber wie kann man diesen Anspruch in einem einzigen Band erfüllen? Die Autoren setzten sich zum Ziel, ein Werk für Vollprofis und totale Laien zu schreiben, die den Computer vielleicht noch gar nicht besitzen. Dies ist ihnen leider nicht gelungen. Was da als Kapitel für Einsteiger deklariert wird, nützt nur demjenigen, der den Computer auch wirklich besitzt und GEM in natura vor sich sieht. Der aber benötigt es nicht, denn das meiste kann auch im beiliegenden Handbuch nachgelesen werden.

Vergessen wir also die ersten 70 Seiten und wenden uns dem großen Abschnitt "GEM aus Programmierersicht" zu. Programmierer und vor allem die C-Tüftler werden mit der Einbindung von GEM-Routinen in eigene Programme vertraut gemacht und in die Lage versetzt, die Benutzeroberfläche für eigene Zwecke zu verwenden. Hinzu kommt eine Einführung in die Anwendung des Entwicklungspakets von Atari und eine Auflistung aller GEM-Funktionen mit ausführlicher Erklärung, dem Aufruf in C, Datentypen sowie Ein- und Ausgabeparametern. Zum Schluß gehen die Autoren noch auf das Hard- und Softwarekonzept des Atari ST ein.

Das vorliegende Buch ist für C-Anwender interessant, die GEM in eigenen Programmen nutzen wollen und es auch wohl tun müssen, um adäquate Software zu erstellen. Einsteigern, die eventuell noch vor der Kaufentscheidung stehen, ist es aber nicht zu empfehlen. Ansonsten ist das Buch hervorragend geschrieben und für den Programmierer sehr informativ.

Thomas Kern

Strategiespiele und wie man sie auf dem Atari 600/800 XL programmiert

Von Schneider
Verlag Data Becker
180 Seiten, 29,– DM
ISBN 3-89011-077-0

Strategiespiele üben auf viele Computerbesitzer einen besonderen Reiz aus. Schnell kommt da der Wunsch auf, selbst einmal ein solches Programm zu erstellen. Leider ist dies aber



gar nicht so einfach. Schon ein halbwegs lauffähiges und dabei noch interessantes Programm erfordert viel Zeit und Arbeit.

Hilfestellung verspricht hier das vorliegende Buch. Wer nun allerdings erwartet, dort nachlesen zu können, wie kriegerische Simulationen oder strategische Weltraumkämpfe programmiert werden, wird wohl enttäuscht. Vielmehr sind es Brettspiele wie Dame, Mühle und Schach, die der Autor zur Veranschaulichung der Materie heranzieht. Einige Listings verdeutlichen die theoretischen Ausführungen. Auch wenn der Leser auf den ersten Blick nicht das vorfindet, was er vielleicht erwartet hat, so bekommt er doch einen Einblick in Abläufe, die für alle Strategiespiele notwendig sind. Erklärungen, Listings, Grafiken usw. erleichtern den Einstieg erheblich. Allen, die einmal einen Blick hinter die Kulissen dieser Programmiertechnik werfen wollen, kann dieses Buch empfohlen werden.

Stephan König

30 Basic-Programme für den Atari

Von Tom Rugg u. a.
Verlag IDEA
170 Seiten, 34,– DM
ISBN 3-88793-032-0

Programmsammlungen in Buchform vorzustellen, ist immer etwas problematisch. Man kann ja als Rezensent nicht alle Listings eintippen und auf Lauffähigkeit und Wert für den Anwender überprüfen. Andererseits sollten aber auch nicht nur die Angaben des Verlags wiedergegeben werden. Außerdem ist ein Werturteil sehr relativ; was dem einen gefällt, findet der andere langweilig.

So verhält es sich auch mit dem hier vorgestellten Werk. Auf rund 170 Seiten haben mehrere Autoren 32 Programme zusammengetragen, die auf allen "kleinen" Atari-Computern (400, 800, 600 XL, 800 XL usw.) laufen und damit eine große Gemeinde ansprechen. Die Programmsammlung teilt sich in mehrere Bereiche auf, um die verschiedensten Anwendungen vom Spiel bis zum Mathematikprogramm abzudecken. Positiv finde ich, daß die einzelnen Listings nicht kommentarlos abgedruckt wurden. Jedes Programm wurde mit Informationen versehen, die wie folgt aufgebaut sind: Zweckbeschreibung – Gebrauchsanweisung – Listing – Einfache Änderungen – Programmaufbau – Variablenliste.

Damit kann jeder etwas anfangen und hat neben der reinen Programmsammlung noch einen Lerneffekt zu verzeichnen. Meiner Meinung nach eignet sich dieses Buch besonders für Einsteiger, die nur wenige Programme besitzen und ihre Bibliothek erweitern wollen. Abschließend eine Aufstellung der einzelnen Titel:

- Biorhythmus
- Entscheidung
- Darlehensabrechnung



- Benzinverbrauch
- Stoppuhr
- Mathe
- Gedächtnistraining
- Zahlen lernen / Vokabeln
- Entziffern
- Würfelspiel
- Geheimwort
- Hindernisspiel
- Autorennen
- Wari
- Kaleidoskop
- Funken
- Vierecke
- Artisten
- Kurve / Differential / Graph / Integral
- Lineare Gleichungen / Statistik
- Geburtstagsberechnung
- Pi / Potenzen / Pythagoras

Rolf Knorre

Der Atari ST

Von Irene und Peter Lüke
Verlag Markt & Technik
200 Seiten, 49,– DM
ISBN 3-89090-229-4

Bücher vom Verlag Markt & Technik finden bei mir besonderes Interesse, da sie immer von kompetenten Autoren geschrieben werden. So war es



auch beim vorliegenden Buch in der 2. überarbeiteten Auflage.

Diesmal bin ich aber nicht recht glücklich geworden. Geschrieben wurde das Buch für ST-Einsteiger, die mit ihrem Computer-Handbuch verständlicherweise nicht zufrieden sind. So nimmt denn auch die Beschreibung der vorhandenen Hardware mit ihren Möglichkeiten einen erheblichen Teil ein. Was mich aber irritiert, ist die Tatsache, daß die Autoren schon zu Anfang (z. B. bei der Beschreibung des Systemaufbaus) ans Eingemachte gehen. Da kommen dann gleich Begriffe wie asynchrone Bussteuerung, Interrupt-Maske oder Supervisor-Stackpointer vor. Der Neuling ist damit sicher überfordert. Hier wäre weniger mehr gewesen; ein grober Überblick hätte gereicht.

Abgesehen von dieser Überladung ist das Buch eine Bereicherung für den Anwender. Es bietet doch erheblich mehr Informationen, als man sie von Atari selbst bekommt. Erst dadurch kann man die Möglichkeiten des ST erkennen. Gutes Papier, sauberer Druck und Hardcover verstärken den positiven Eindruck. Wer den relativ hohen Preis nicht scheut, sollte sich das Buch ruhig zulegen.

Rolf Knorre

Logo auf dem Atari ST

Von Dieter und Jürgen Geiß
Hüthig Verlag Heidelberg
150 Seiten, 35,– DM
ISBN 3-7785-1262-5

Logo hat neben Basic und C nach wie vor einen festen Platz in der Hitliste der Programmiersprachen für den Atari ST. Das beiliegende Handbuch ist zwar in Deutsch geschrieben, doch läßt es die nötige Sorgfalt vermissen. Wer sich intensiv mit dieser Sprache beschäftigen möchte, benötigt mehr und vor allem bessere Informationen. Dies verspricht nun das vorliegende Buch und hält es auch.

Neben der grundsätzlichen Bedienung des Editors und der Arbeit mit dem Wortschatz von Logo gehen die Autoren auch auf kompliziertere Sachverhalte ein und beschreiben fortgeschrittene Programmiertechniken. Wo das Handbuch gut dazu geeignet war, sich mit den Fähigkeiten des ST-Logo vertraut zu machen, hilft dieses Buch nun, dies auch wirklich zu nutzen.

Wort- und Listenverarbeitung, Fehlerbehandlung und Debugging, Diskettenverwaltung und viele andere Themen werden von den Autoren behandelt. Dabei gehen sie auch auf die grundsätzliche Methode des Programmentwurfs ein und bewahren den Leser davor, einen sogenannten Spaghetticode zu entwerfen. Er lernt auf diese Weise, die Vorteile der strukturierten Programmierung in Logo konsequent zu nutzen.

Glücklicherweise bleibt dem Leser ein erneuter Ausdruck der Befehlsübersicht aus dem Handbuch erspart. Die Autoren beschränken sich auf einen kurzen, knappen, aber nützlichen Anhang, in dem die Fehlermeldungen, die Steuerzeichen und der Grundwortschatz dargestellt werden.

Wer auf dem Atari ST in Logo programmieren möchte, sollte sich dieses Buch anschaffen. Im Gegensatz zu ähnlichen Werken werden nicht Wiederholungen aus dem Handbuch geboten, sondern wirkliche Erweiterungen.

Thomas Kern

Was der Atari alles kann (Band 1 + 2)

Von Alma und Johann Peschetz
Vogel Buchverlag
ca. 400 Seiten, 35,– DM
ISBN 3-8023-0795-X

Die vorliegenden Bände wollen Besitzern eines Atari 600/800 XL/130 zeigen, was mit ihrem Computer alles möglich ist. Auf insgesamt über 400 Seiten haben die Autoren dann auch so viel Material zusammengetragen, daß der Anfänger vielleicht schon wieder erschreckt wird. Doch ist dies kein Grund zur Panik, denn durch die vielen Beispielprogramme ist immer dafür gesorgt, daß der Leser nicht nur graue Theorie vorfindet.

Band 1 bietet eine grundlegende Einführung in die Programmierung von Grafik, Sound und all dem, was der Atari sonst noch kann. Band 2 setzt dann schon einige Kenntnisse voraus, ist aber immer noch leicht verständlich. Wer sich gerade einen "kleinen" Atari zugelegt hat und mit dem Handbuch verständlicherweise



nicht zurechtkommt, sollte sich diese beiden Werke kaufen. Fortgeschrittene werden allerdings nicht allzuviel Neues finden.

Stephan König

Atari ST – Peeks & Pokes

Von Stefan Dittrich
Verlag Data Becker
208 Seiten, 29,– DM
ISBN 3-89011-148-3

Dieses Buch wurde schon kurz nach Erscheinen der ST-Computer auf den Markt gebracht und liegt jetzt in einer überarbeiteten Auflage vor. Der Autor beschäftigt sich intensiv mit den internen Abläufen des Computers und den Eingriffen, die der Benutzer relativ einfach vornehmen kann. Leider ist die Überarbeitung scheinbar schon vor einiger Zeit erfolgt, da immer noch davon ausgegangen wird, daß das Betriebssystem von Diskette geladen werden muß. Mittlerweile sind bekanntlich ROMs zu haben und auch schon weit verbreitet. Wenn man sich für dieses Buch, das viele gute Informationen bietet, interessiert, sollte man die nächste Überarbeitung abwarten.

Stephan König



3 Major Motion

Schreckliches geht vor sich auf den Highways der Stadt. Eine Gruppe, die sich selbst Draconian League nennt, terrorisiert friedliche Autofahrer. Glücklicherweise ist der Spieler per Zufall vor Ort und hat auch noch ein neues Wunderauto zur Verfügung, um dem Schrecken ein Ende zu machen.

Das ist die komprimierte Rahmenhandlung zu "Major Motion", dem neuen Spiel aus dem Hause Microdeal. Wie so oft kann man dieser Story kaum entnehmen, um was für ein Spiel es sich eigentlich handelt. Dabei ist das Programm recht simpel. Auf dem Monitor sieht man eine Straße aus der Vogelperspektive, auf der reger Verkehr herrscht. Nach Programmstart taucht ein LKW auf, der automatisch die Heckklappe öffnet und das Wunderauto während der Fahrt entlädt. Jetzt ist es am Spieler, möglichst unbeschadet auf der Fahrbahn zu bleiben und dabei noch die Gegner, die Mitglieder der Draconian League, zu erledigen.

Bedient wird der Wagen mit der Maus. Man kann nach links oder rechts fahren, beschleunigen, bremsen und natürlich schießen. Es stehen zahlreiche Waffen zur Verfügung, die aber nicht sofort einsatzbereit sind.

Die gestellte Aufgabe ist meiner Meinung nach sehr schwierig. Da die Gegner nicht nur mit Wagen, sondern vereinzelt auch mit Hubschraubern angreifen, ist das Überleben reine Glückssache (oder nur mit viel Übung möglich). Die Grafik des Programms ist nicht schlecht, aber auf Dauer wenig abwechslungsreich. Mal sieht man nur eine Fahrbahn, mal wird die Straße geteilt.

Wer sich gerne mit schwierigen Aufgaben beschäftigt, die im Grunde aber doch nur ein Schießspiel darstellen, sollte "Major Motion" einmal ausprobieren.



Nicht sehr anregend: das waffenstarrende Wunderauto von oben betrachtet

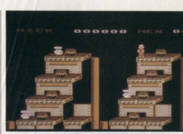
System: Atari ST
Hersteller: Microdeal

Rolf Knorre

3 Nuclear Nick



Das britische Softwarehaus Americana bemüht sich ständig, die Benutzer der Atari XL/XE ausreichend mit Software zu versorgen. Eines seiner neuen Programme ist "Nuclear Nick". Es handelt sich dabei um ein reines Actionspiel, bei dem Schnelligkeit und Geschicklichkeit eindeutig im Vordergrund stehen.



Der nukleare Nick wird von wildgewordenen Robotern gejagt

In 20 verschiedenen Räumen tummeln sich zahlreiche Roboter, deren CPU scheinbar ein wenig durcheinandergeraten ist. Dadurch sind sie zu einer echten Gefahr geworden, die der nukleare Nick bannen soll. Dazu muß er durch die Räume rennen, Powerpacks berühren, um die nötige Energie zu gewinnen und dann einfach auf die Roboter losstürmen. Durch die Powerpacks ist er nämlich fast unverwundbar geworden und kann die Maschinen auflösen. Ohne diese Ladung ist er völlig hilflos und verliert bei Berührung mit einem Roboter eines der vier Leben. Jeder Raum ist in verschiedene Etagen aufgeteilt. Fällt Nick zwei Etagen tief, hat er wieder ein Leben eingebüßt.

Schön ist bei diesem Spiel, daß man über die SELECT-Taste einen der 20 Räume auswählen kann. Man muß sich also nicht unbedingt der Reihe nach durcharbeiten. Außerdem ist es so möglich, an manchen Stellen ein bißchen zu üben.

Eine neue Spielidee bietet "Nuclear Nick" sicher nicht, dafür aber ein hohes Tempo, eine ansehnliche Grafik und viel Spaß.

System: Atari XL/XE
Hersteller: Americana
Bezugsquelle: Compy Shop

Stephan König



4

Beer Belly Burt's Brew Biz

Wenn ein Programm einen so witzigen Titel hat (damit kann man den Verkäufer im Laden zu Verzweiflung bringen), erwartet man automatisch auch etwas Besonderes. Leider wird man hier aber enttäuscht.

Der Covertext verrät fast nichts. Er besagt nur, daß Burt eine Brauerei besichtigen soll, die aus 30 Räumen besteht. Bei Spielstart steht der kleine Burt vor drei Türen, die jeweils zu verschiedenen Abteilungen führen. Hat man eine davon durchschritten, wechselt das Bild. Was dann auftaucht, ist mit den unzähligen, bereits vorhandenen Plattformspielen zu vergleichen.



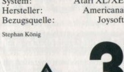
Eine wirkliche Brauereibesichtigung ist sicher berauschender als dieses Spiel

Was Burt nun genau tun soll, bleibt unklar. Es sind keine Gegenstände aufzunehmen und keine Geheimnisse zu entschlüsseln. Der einzige Gegner, der auftaucht, sieht aus wie Burt und schießt manchmal. In der Regel ist man aber schneller und besiegt ihn.

Ich halte vorliegendes Programmm für eine echte Enttäuschung. Sollte jemand herausfinden, was in diesem Spiel eigentlich zu tun ist, kann er es uns vielleicht verraten.

System:	Atari XL/XE
Hersteller:	Americana
Bezugsquelle:	Joysoft

Stephan König



3

Space-Station

Nun erobern schon die Action-Adventures den Atari ST. Das erste erwähnenswerte Spiel dieser Art heißt "Space Station" und besteht aus mehr als 100 Bildern, die viele trickreiche Aufgaben und Gefahren enthalten.

Mit dem Joystick steuern Sie einen todesmutigen Raumschiffkommandanten, der sich in eine von Außerirdischen besetzte Raumstation wagt. Nachdem er die Situation erfaßt hat, gibt es für ihn nur noch zwei Ziele: Die Station muß zerstört werden, damit sie nicht in die Hände der Aliens fällt. Außerdem muß sein defektes Raumschiff mit Treibstoff und Energie versorgt werden, um die Rückkehr zur Erde zu sichern.

Sie gelangen nun in das Labyrinth der Raumstation, in der viele Gefahren lauern. Hier müssen Sie die verstreut herumliegenden Lithiumkristalle aufsameln, um das Raumschiff zu versorgen, und die Diskette finden, die den Selbstzerstörungsmechanismus der Station aktiviert. Ihr eigener Energievorrat ist zu Beginn noch relativ groß, aber in den Kämpfen mit den Aliens und anderen bösen Fallen schwindet er mehr und mehr. Zur Verteidigung steht Ihnen eine Waffe zur Verfügung, zu der Sie allerdings noch Munitionspakete finden müssen. Diese sind aber nur mäßig bestückt, was zum sparsamen Umgang mit der Pistole zwingt. Wer Glück hat, findet auch noch einen Raketen-Rucksack, mit dem man über Gefahren hinwegfliegen kann. "Space Station" ist ein interessantes Action-Adventure, das die Reihe dieser Programme auf kleineren Rechnern nahtlos fortführt. Fans derartiger Spiele werden sicherlich begeistert sein.

Grafisch bietet "Space Station" den üblichen Standard. Was den Sound betrifft, hört man nur Gedudel über den Lautsprecher. Erst bei Anschluß eines Midi-Synthesizers kommt man in den vollen Genuß des Klangs.

System:	Atari ST (512 KByte, Farbmonitor)
Hersteller:	Eidersoft
Bezugsquelle:	Profisoft

Thomas Kern



Das Innere der Raumstation erweist sich erwartungsgemäß als Labyrinth

ST-Protector

Man nehme eine alte, bewährte Spielidee und setze sie mit ein paar Veränderungen auf einen neuen Rechner um. Nach diesem Prinzip arbeiten heute einige Hersteller und Programmierer, um der Softwareflaute entgegenzuwirken. Daß darunter manchmal die Qualität leidet, ist eigentlich selbstverständlich.



3

Bei "ST-Protector" stand der Klassiker "Defender" Pate. Ein Raumschiff düst über eine Planetenlandschaft und muß sich gegen die einfallenden Aliens verteidigen. Im oberen Viertel des Bildschirms zeigt das Radar die herannahenden Gegner an. In brenzligen Situationen kann man eine Superbombe zünden oder durch Hypersprung entkommen.

Soweit ist alles schon einmal dagewesen. Hinzu kommt bei "ST-Protector" nun die Rettung der Bewohner eines Planeten. Hilflos winkend stehen sie da, und nur der kühne Streiter mit seinem silberblitzenden Raumschiff kann ihnen zu Hilfe eilen. Trotz dieser Neuerung entspricht das Spiel im großen und ganzen dem Programm "Defender".

Gesteuert wird mit dem Joystick, nicht – wie beim Atari ST üblich – mit der Maus. Für Hypersprung und Smartbombe muß man die Tastatur zu Hilfe nehmen.

Die Grafik ist gerade das mindeste, was man einem Spiel auf dem Atari ST erwarten kann. Mit ein wenig mehr Mühe hätte man sie auch besser gestalten können. Die Frage ist allerdings, ob das bei einem solchen Spiel notwendig ist. Sehr gut ist jedoch der Sound. Natürlich erreicht man über den Lautsprecher nur bei deren C-64-Standard, wobei selbst das Gerät mit 8 Bit noch mehr bieten könnte. Den vollen Genuß bringt erst der Anschluß eines Midi-Synthesizers, der die wahre Klangfülle des Programms offenbart. Da nur wenige über ein solch kostspieliges Gerät verfügen, stellt sich die Frage, ob man nicht besser Mühe auf den reinen ST-Sound verwendet hätte.

"ST-Protector" ist ein empfehlenswertes Actionspiel für alle, die bisher vergeblich auf ein vernünftiges Programm gewartet haben. Sicherlich wurden die Möglichkeiten des Computers noch nicht voll ausgeschöpft, aber der Spielefreak wird mit dem Ergebnis zufrieden sein.

System:	Atari ST (512 KByte, Farbmonitor)
Hersteller:	Eidersoft
Bezugsquelle:	Profisoft

Thomas Kern



2

Polar Pierre

Nicht mehr ganz neu aber doch sehr gut ist das Programm "Polar Pierre". Es bietet dem Spieler eine Menge guter Unterhaltung.

Die Aufgabe ist einfach zu beschreiben. Der Spieler muß Pierre durch eine Anzahl von Screens steuern. In jedem Bild tauchen einige Fähnchen auf, die Pierre berühren muß, um sie zu aktivieren. Erst wenn alle Fahnen eines Bildes aktiv sind, kommt man ins nächste. So einfach, wie es scheint, ist diese Aufgabe aber nicht.

Schon im ersten Bild fängt der Ärger an. Ein fürchterliches Gewitter tobt (der Sound ist beeindruckend). Das Unwetter alleine kann Pierre nicht erschrecken, wohl aber die energiegeladenen Blitze. Viel zu schnell wird Pierre davon getroffen und haucht dann eines seiner vier Leben aus. In jedem neuen Bild tauchen andere Gefahren auf. Da die Fahnen auf mehrere Plattformen verteilt sind, ist es fast immer die letzte, an die man nur schlecht herankommt. Wenn man Pierre dann noch über vereiste Flächen steuern muß, auf denen der kleine Kerl ständig ausrutscht, hat man alle Hände voll zu tun.



Das Spiel kann um eigene Screens ergänzt werden

Grafik und Spielaufbau, aber auch die Spielidee gefallen mir sehr gut. Positiv ist, daß man sich eigene Räume zusammenstellen kann. Dazu dient die Construction Factory. Leider können nur die Besitzer eines Diskettenlaufwerks die neuen Screens auch abspeichern und wiederverwenden. Ich halte "Polar Pierre" für ein witziges Spiel und eine Bereicherung des Softwaremarktes.

System:	Atari 400/800/600 + 800 XL/130 XE
Hersteller:	Databyte
Bezugsquelle:	Microhändler

Rolf Knorre

Vielleicht haben Sie es sich schon gedacht. Die großen Ziffern stehen für die Noten, die wir für die jeweiligen Spiele vergeben haben: 1 = sehr gut, 6 = ungenügend.



4

Screaming Wings

Als ich das erste Mal die Casette von "Screaming Wings" in Händen hielt, dachte ich, es müsse sich wohl um eine Art Flugsimulation handeln. Ich hatte mich allerdings gewaltig getäuscht. Das Programm ist eher ein klassisches Schießspiel nach Invader-Muster. Zwar erhielt es ein moderneres Outfit, die Ähnlichkeit ist aber trotzdem nicht zu übersehen.



Gegen den Pulk von Angreifern hilft wieder einmal nur Schießen

Es geht um turbulente Luftkämpfe, die der Spieler mit seinem Flugzeug bestehen muß. Dieses schwebt am unteren Bildschirmrand und kann nach links oder rechts und ein Stückchen nach oben bewegt werden. Von den Seiten und von oben tauchen nun nach und nach verschiedene Angreifer auf, mal in Wellen, mal als Einzelkämpfer. Die Aufgabe ist klar: schießen, bis die Rohre heiß werden und dabei möglichst viele Punkte machen.

Einige der gegnerischen Flugzeuge verwandeln sich nach dem Abschuß in Buchstaben, die aufgefangen werden müssen. Ein X verstärkt die Feuerkraft, ein D bringt die wichtige Destruction-Bombe. Nur mit ihr kann das große Kampfflugzeug vernichtet werden. Ohne diese Bombe bedeutet sein Auftauchen fast immer den Verlust eines Lebens, von denen 3 zur Verfügung stehen. Bei 10.000 Punkten bekommt man ein Extraleben.

Leider ist die Grafik nicht überragend. Da auch die Handlung weder neu noch originell ist, kann "Screaming Wings" nur großen Fans von Schießspielen empfohlen werden.

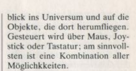
System: Atari 800 XL/130 XE
Hersteller: Red Rat Software
Bezugsquelle: Microhändler

Rolf Knorre

Pygnosis – Software aus England

Brandneue Software bekamen wir dieser Tage aus England. Der Name Pygnosis ist Besitzern eines Atari ST ja sicher nicht mehr unbekannt. Wer die beiden neuen Spiele dieser Firma kennt, wird ihr vielleicht auch eine tolle Zukunft prophezeien.

Nun aber zu den Spielen. Voraussetzung für "Deep Space" und "Arena" sind ein ST mit Farbmonitor und mindestens 512 KByte Speicherplatz. Bei "Deep Space" handelt es sich um ein intergalaktisches Weltraumabenteuer. Mit dem hochmodernen Strix-Fighter kann der Spieler verschiedene Sonnensysteme erkunden. Vom Raumschiff selbst sieht man nur die Kontrollsysteme am oberen und unteren Bildschirmrand. In der Mitte des Bildes hat man einen prima Ausblick ins Universum und auf die Objekte, die dort herumfliegen. Gesteuert wird über Maus, Joystick oder Tastatur; am sinnvollsten ist eine Kombination aller Möglichkeiten.



Die Kontrollinstrumente und Statusanzeigen des Bordcomputers sind vielfältig. Vom Zustandsbericht über Radaranzeigen bis zu den verschiedenen Waffensystemen kann alles aufgerufen werden. Wenn man den Raum verdunkelt und sich voll auf "Deep Space" konzentriert, kommt schon bald ein echtes Science-fiction-Gefühl auf. Je nach Lage der Dinge kann man recht schnell in eine Weltraumschlacht verwickelt werden, bei der dann Quarkbombs oder Photonentorpedos zum Einsatz kommen; manchmal muß man aber auch einem Meteorsturm oder einer Sonne ausweichen.



Pygnosis-Startscreen

Die Grafik des Spiels ist recht gut, obwohl noch lange nicht alle Möglichkeiten des ST genutzt wurden. Das gilt auch für den Sound. Einfach fantastisch sind aber die Grafiken vor Spielbeginn und zwischen zwei Runden. Wer Schießspiele und technische Programme mag, wird an "Deep Space" sicher Gefallen finden.



Mehr Aktivität als bei der Sportschau wird hier vom Sportbegeisterten schon verlangt



Nur mit Farbmonitor wird dieser Ausblick in die Tiefe des Raums gewährt

Einen größeren Anhängerkreis wird wohl das zweite Pygnosis-Programm finden. Es trägt den Titel "Arena". Dahinter verbirgt sich aber kein altrömischer Gladiatorenkampf, sondern ein neuzeitliches Sportspiel der allerersten Güte. Insgesamt können bis zu vier Personen in sechs Sportarten teilnehmen. Es geht zwar auch alleine, doch macht es in Gesellschaft den meisten Spaß. In folgenden Disziplinen kann man antreten: 100-Meter-Lauf, Weitsprung, Hochsprung, Stabhochsprung, Kugelstoßen, Speerwurf. Alle Sportarten sind somit der Leichtathletik zuzurechnen, was aber nicht bedeutet, daß man leichtes Spiel hätte. Die Aktionen werden über die Tastatur gesteuert. Wie man es schon von anderen Spielen dieser Art kennt, muß z.B. die Energie für den Anlauf durch abwechselndes Drücken zweier Tasten gesteigert werden, was allerdings sanft formuliert ist; man muß geradezu draufhämmern.

Bei "Arena" handelt es sich um ein recht unkompliziertes Programm. Zum Ausgleich erhält man eine tolle Grafik und gute Unterhaltung. Der zu steuernde Sportler ist, wie auch sein Umfeld, hervorragend dargestellt. Ist ein Lauf oder Sprung beendet, erscheint auf dem Bildschirm ein Männlein mit dicker Nase, das wohl eine Art Schiedsrichter sein soll. Zur Freude der Spieler entwickelt sich dann immer ein kleiner Dialog zwischen den beiden. Die Texte erscheinen in Sprechblasenform und sind teilweise recht ulkig. Da man jederzeit freien Zugriff auf eine gewünschte Sportart hat, kann man besonders schwierige Disziplinen öfter anwählen. Leider gibt es nach erfolgreicher Bewältigung der Sportarten keine besondere Wertung. Man kann aber zu jeder Zeit eine Punktetabelle auf den Bildschirm holen, um den aktuellen Stand zu überprüfen. Dieser kann auch abgespeichert werden, um das Spiel zu einem späteren Zeitpunkt fortzusetzen. "Arena" ist ein gelungenes Programm, wenn man einmal vom bescheidenen Sound absieht.

System: Atari ST (min. 512 KByte, Farbmonitor)
Hersteller: Pygnosis
Bezugsquelle: Joysoft

Rolf Knorre

A Day at the Races

Aus der Menge der üblichen Actionspiele ragt dieses Programm etwas heraus, da es weniger um schnelle Aktionen als um Glück geht. Wie der Titel schon verrät, dreht es sich um Rennen, speziell um Pferderennen.

Der Spieler kann sich aus fünf Pferden eines aussuchen, auf das er wettet. Danach startet das Rennen, das auf dem Bildschirm in Bewegung umgesetzt wird. Am Ende gibt es eine Auswertung. Je nachdem, welchen Platz das gewählte Pferd belegt, hat man Geld gewonnen oder verloren. Insgesamt werden 10 solcher

4

Rennen durchgeführt, bevor die Endabrechnung erfolgt.

Leider sind die Möglichkeiten, die der Spieler zum Eingreifen hat, sehr gering. Man muß sich also auf das Beobachten beschränken, was auf Dauer doch unbefriedigend ist.

System: Atari 400/800 XL/XE
Hersteller: Red Rat Software
Bezugsquelle: Compy Shop

Stephan König

Pferderennen wie im richtigen Leben: Wette abschließen und abwarten

INSERENTEN

Anzeigenschluß für die Februar/März-Ausgabe ist der 20.1.1987.

Falls Fragen auftauchen, steht Ihnen Herr Arno Weiß gerne zur Verfügung.
☎ 07252/3058

VORSCHAU

Antic

Den ganz eingefleischten Atarianern ist sie bekannt, die Konkurrenz des **ATARI** ***magazins*** jenseits des Atlantiks. Was dem Spekulanten das Wallstreet Journal, ist dem ST-User Antic, die amerikanische Zeitschrift für Atari ST. Ein Mitarbeiter hat bei den amerikanischen Kollegen ein bißchen spioniert. Was er herausgefunden hat, steht im nächsten Heft.

320 KiloByte

Der Besitzer eines "kleinen" 800 XL braucht sich beileibe nicht vor den "größeren" Atari-Computern zu verstecken. Denn wer weiß besser als er, welche Qualitäten in dieser kleinen Maschine stecken! Aber gelegentlich stößt man eben doch an die 64-KByte-Grenze und hätte gerne ein größeres RAM.

Wie Sie mit etwas Geschick und kaum mehr als 80 DM zu sagenhaften 320 KByte kommen, lesen Sie im nächsten **ATARI** ***magazin***. Eine ausführliche Bauanleitung mit Schaltplänen und der notwendigen Software macht Ihnen die Anwendung leicht.

Sound

Der im Atari ST eingebaute Soundchip steht in einem krassen Mißverhältnis zur sonstigen Ausstattung der Maschine; seine Leistungen sind eher kläglich. Über die MIDI-Schnittstelle kann aber mit einem entsprechenden Synthesizer ganz ordentlich musiziert werden. Notenschreiben, Komponieren und vieles mehr übernimmt der "Ministrel Musical Editor". Mehr darüber in Heft 2/87 des **ATARI** ***magazins***.

Das nächste ATARI *magazin* erscheint am 18.2.87

IMPRESSUM

Herausgeber
Werner Rätz
Dipl.-Wirt.-Ing. (FH) Thomas Eberle

Chefredakteur
Thomas Eberle

Technische Redaktion
Werner Rätz

Redaktion
Helmut Fischer
Robert Kaltenbrunn

Ständige freie Mitarbeiter
Rolf Knorre
Thomas Tai
Dipl.-Ing. Peter Finzel
Thomas Tausend

Versandservice
Gabriele Herzog

Titelbild
Rainer Grinda

Anzeigen
Arno Weiß
Es gilt die Anzeigenpreisliste Nr. 1

Montage
Frederique Melchers

Satz
Druckerei Sprenger
7143 Vaihingen/Enz

Druck
Südd. Zeitungsdienst GmbH
Druckerei- u. Verlags-GmbH
7080 Aalen

Vertrieb
Verlagsunion
6200 Wiesbaden

Anschrift des Verlags
Verlag Rätz-Eberle
Postfach 1640
Melanchthonstraße 75/1
7518 Bretten
Telefon 07252/3058

Das **ATARI** ***magazin*** erscheint zweimonatlich jeweils zur Mitte des Vormonats.
Das Einzelheft kostet 6.– DM.